



# DISTINGUISHING CONVERGENCE ON PHYLOGENETIC NETWORKS

Submitted in fulfilment of the requirements of the degree

of

Doctor of Philosophy

in

Mathematics

at

University of Tasmania

Faculty of Science, Engineering & Technology

School of Physical Sciences

Discipline of Maths

Jonathan Mitchell, BSc (Hons) *UTAS*

June 2016



## Declaration of Originality

This thesis contains no material which has been accepted for a degree or diploma by the University or any other institution, except by way of background information and duly acknowledged in the thesis, and to the best of my knowledge and belief no material previously published or written by another person except where due acknowledgement is made in the text of the thesis, nor does the thesis contain any material that infringes copyright.

Signature

\_\_\_\_\_ Date

21/06/2016



## Authority of Access

This thesis may be made available for loan and limited copying and communication in accordance with the Copyright Act 1968.

Signature

\_\_\_\_\_ Date

21/06/2016



## Abstract

In phylogenetics, the evolutionary history of a group of taxa, for example, groups of species, genera or subspecies, can be modelled using a phylogenetic tree. Alternatively, we can model evolutionary history with a phylogenetic network. On phylogenetic networks, edges that have previously evolved independently from a common ancestor may subsequently converge for a period of time. Examples of processes in biology that are better represented by networks than trees are hybridisation, horizontal gene transfer and recombination.

Molecular phylogenetics uses information in biological sequences, for example, sequences of DNA nucleotides, to infer a phylogenetic tree or network. This requires models of character substitution. A group of these models is called the *Abelian group-based models*. The rate matrices of the Abelian group-based models can be diagonalised in a process often referred to as *Hadamard conjugation* in the literature. The time dependent probability distributions representing the probabilities of each combination of states across all taxa at any site in the sequence are referred to as *phylogenetic tensors*. The phylogenetic tensors representing a given tree or network can be expressed in the diagonalised basis that may allow them to be analysed more easily. We look at the diagonalising matrices of various Abelian group-based models in this thesis.

We compare the phylogenetic tensors for various trees and networks for two, three and four taxa. If the probability spaces between one tree or network and another are not identical then there will be phylogenetic tensors that could have arisen on one but not the other. We call these two trees or networks *distinguishable* from each other. We show that for the binary symmetric model there are no two-taxon trees and networks that are distinguishable from each other, however there are three-taxon trees and networks that are distinguishable from each other.

We compare the time parameters for the phylogenetic tensors for various taxon label permutations on a given tree or network. If the time parameters on one taxon label permutation in terms of the other taxon label permutation are all non-negative then we say that the two taxon label permutations are not *network identifiable* from each other. We show that some taxon label permutations are network identifiable from each other.

We show that some four-taxon networks do not satisfy the four-point condition, while others do. There are two “structures” of four-taxon rooted trees. One of these structures is defined by the cluster, b,c,d, where the taxa are labelled alphabetically from left to right, starting with a. The network with this structure and convergence between the two taxa with the root as their most recent common ancestor satisfies the four-point condition.

The phylogenetic tensors contain polynomial equations that cannot be easily solved for four-taxon or higher trees or networks. We show how methods from algebraic geometry, such as Gröbner bases, can be used to solve the polynomial equations. We show that some four-taxon trees and networks can be distinguished from each other.



## Acknowledgements

I would like to thank my supervisor, Barbara Holland, and my co-supervisor, Jeremy Sumner, for their wonderful support over the course of my PhD. I am very appreciative of everything they have done for me. They have taught me almost everything I know about phylogenetics. Regular meetings often contained insightful discussions that lead me to make discoveries that I may not have made otherwise. I would not have been able to pursue a PhD in the topic I chose if it weren't for the previous work of Jeremy, Barbara and their collaborators, in particular Peter Jarvis. The help they have given me on writing my thesis has been extremely constructive. I am also very grateful for their support of my wish to take some time off and suspend my candidature. If it were not for this then I may not have completed my PhD.

Barbara's knowledge of phylogenetics is incredibly broad. She is a leader in the field of phylogenetics and brings together researchers from various disciplines, which is vital in phylogenetics. This is reflected in her many collaborators in the biological fields, for which she provides statistical and mathematical expertise. Barbara almost invariably had the answer when I had a general question about phylogenetics or a question about statistics. It has been Barbara's reassurance and knowledge of what constitutes a PhD thesis that has given me confidence that my work is of the standard of a PhD.

Despite being officially a co-supervisor, Jeremy has provided me help on the level of a main supervisor. Most of our meetings were in Jeremy's office with both Jeremy and Barbara. Jeremy's skills in algebra are excellent. He played a major part in developing the convergence-divergence networks for which my thesis relies on. Jeremy was of great assistance when I had a question about algebra and how it related to the convergence-divergence networks.

I would also like to thank Amelia Taylor, who specialises in computational commutative algebra at Colorado College. I had the idea of using techniques from algebraic geometry, eg. using Gröbner bases, to solve the polynomial equations that I was dealing with. While visiting UTAS, Amelia helped me to understand the necessary algebraic geometric concepts and helped me to install and use Macaulay2, a computer algebra system. Macaulay2 was required for a large part of Chapter 5.

I would like to thank Michael Brideson, the Graduate Research Coordinator in Mathematics, and Karen Bradford, the Executive Officer of the Maths and Physics department, for their general help during my PhD.

I would like to thank my family, particularly my parents, Bron and Chris, and my sister,

Emma, for their continued love and belief in me.

I would like to thank my wonderful partner, Ella, who has been an amazing addition to my life.

I would like to thank the University of Tasmania, in particular the Mathematics and Physics department.

Finally, I would like to thank the Australian Federal Government for providing me with an Australian Postgraduate Award.

# Contents

<b>Declaration of Originality</b>	<b>i</b>
<b>Authority of Access</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>Acknowledgements</b>	<b>vii</b>
<b>1 Introduction &amp; Literature Review</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.1.1 Overview . . . . .	1
1.1.2 Phylogenetics . . . . .	3
1.1.3 Markov Models . . . . .	5
1.1.4 Phylogenetic Tensors on Convergence-Divergence Networks . . . . .	7
1.1.5 Ideals and Gröbner Bases . . . . .	10
1.2 Literature Review . . . . .	10
1.2.1 Networks in Phylogenetics . . . . .	10
1.2.2 Identifiability . . . . .	22
1.2.3 Group-Based Models . . . . .	24
1.2.4 Tensor Representation of Models . . . . .	27
<b>2 Diagonalisation of Markov Models</b>	<b>31</b>
2.1 Markov Chains in Phylogenetics . . . . .	31
2.2 The Markov Property on Transition Matrices . . . . .	32
2.3 Expressing Transition Matrices in Terms of the Rate Matrices . . . . .	33
2.4 Transition Matrices as Matrix Exponentials . . . . .	35
2.5 The Abelian Group-Based Models . . . . .	36
2.5.1 Diagonalising the Abelian Group-Based Models . . . . .	37
2.5.2 The Binary Symmetric Model . . . . .	39
2.5.3 The $\mathbb{Z}_3$ Model . . . . .	41
2.5.4 The K3ST Model . . . . .	43

2.6	Finding Diagonalising Matrices in <i>Mathematica</i> . . . . .	46
2.7	The Four-State General Markov Model . . . . .	47
2.8	Abelian Group-Based Models and Their Diagonalising Matrices . . . . .	48
2.9	Choosing the Diagonalising Matrix Parameters . . . . .	50
<b>3</b>	<b>The Splitting Operator and Convergence</b>	<b>51</b>
3.1	Character Sequence Alignments . . . . .	51
3.2	Tensor and Kronecker Products . . . . .	52
3.3	Phylogenetic Tensors on the General Markov Model . . . . .	52
3.3.1	The Phylogenetic Tensor on the Binary Symmetric Model . . . . .	53
3.4	The Splitting Operator . . . . .	54
3.4.1	Pushing Back the Splitting Operator . . . . .	55
3.4.2	The Splitting Operator on Edges . . . . .	58
3.5	Modelling Convergence . . . . .	60
3.6	Constructing the Trees and Networks . . . . .	61
3.7	Transformed Phylogenetic Tensors in the Hadamard Basis . . . . .	64
3.8	Modelling Convergence on Networks . . . . .	65
<b>4</b>	<b>Two-Taxon and Three-Taxon Trees and Networks</b>	<b>69</b>
4.1	Constraints on the Transformed Phylogenetic Tensors . . . . .	69
4.2	Two-Taxon Phylogenetic Tensors . . . . .	74
4.2.1	Non-Clock-Like Tree . . . . .	74
4.2.2	Clock-Like Tree . . . . .	75
4.2.3	Clock-Like Convergence-Divergence Network . . . . .	76
4.2.4	Network Identifiability . . . . .	77
4.3	Three-Taxon Phylogenetic Tensors . . . . .	79
4.3.1	Non-Clock-Like Tree . . . . .	80
4.3.2	Clock-Like Tree . . . . .	81
4.3.3	Clock-Like Convergence-Divergence Network . . . . .	82
4.3.4	Comparisons of Other Three-Taxon Networks . . . . .	86
4.3.5	Distinguishability of Three-Taxon Trees and Networks . . . . .	88
<b>5</b>	<b>Network Identifiability of Taxon Label Permutations</b>	<b>95</b>
5.1	Permuting the Taxon Labels . . . . .	95
5.2	Determining How Often Time Parameters are Non-Negative . . . . .	97
5.3	The Three-Taxon Clock-Like Tree . . . . .	98
5.4	The Convergence-Divergence Network . . . . .	100
5.5	Pairwise Distances . . . . .	101
5.6	The (12) Taxon Label Permutation . . . . .	106
5.7	The (13) Taxon Label Permutation . . . . .	108

5.8	The (23) Taxon Label Permutation . . . . .	110
5.9	Comparing the (12) Permutation to the (23) Permutation . . . . .	112
5.10	The Six Taxon Label Permutations . . . . .	113
<b>6</b>	<b>Gröbner Bases</b>	<b>115</b>
6.1	Algebraic Geometric Techniques for Finding the Constraints . . . . .	117
6.2	Four-Taxon Convergence-Divergence Networks . . . . .	127
6.2.1	Network 1 . . . . .	130
6.2.2	Network 1 <i>a</i> . . . . .	133
6.2.3	Network 1 <i>b</i> . . . . .	137
6.2.4	Network 1 <i>c</i> . . . . .	139
6.2.5	Network 2 . . . . .	142
6.2.6	Network 2 <i>b</i> . . . . .	143
6.2.7	Network 3 . . . . .	146
6.3	Distinguishability of the Networks . . . . .	148
6.3.1	Pseudocode Algorithm to Distinguish Trees and Networks . . . . .	157
6.4	An Example with Multiple Convergence Periods . . . . .	158
<b>7</b>	<b>Conclusion &amp; Discussion</b>	<b>163</b>
7.1	Thesis Findings . . . . .	163
7.2	Future Work . . . . .	166
	<b>Appendices</b>	<b>171</b>
<b>A</b>	<b>Time Parameters for the (12) Taxon Label Permutation</b>	<b>173</b>
<b>B</b>	<b>Time Parameters for the (13) Taxon Label Permutation</b>	<b>185</b>
<b>C</b>	<b>Time Parameters for the (23) Taxon Label Permutation</b>	<b>195</b>
<b>D</b>	<b>The (23) Permutation in Terms of the (12) Permutation</b>	<b>203</b>
<b>E</b>	<b>Gröbner Basis for Network 1<i>b</i></b>	<b>211</b>
<b>F</b>	<b>Gröbner Basis for Network 1<i>c</i></b>	<b>213</b>
<b>G</b>	<b>Gröbner Basis for Network 2<i>b</i></b>	<b>215</b>
<b>H</b>	<b>Gröbner Basis for Network 3</b>	<b>217</b>
<b>I</b>	<b>Regions in the Probability Space</b>	<b>219</b>
	<b>References</b>	<b>227</b>



# Chapter 1

## Introduction & Literature Review

### 1.1 Introduction

#### 1.1.1 Overview

In this thesis we will discuss a range of topics in phylogenetics and extend upon the current knowledge in the field. The thesis focuses on the analysis of phylogenetic networks, structures that differ from more simple phylogenetic trees and can allow for convergence of multiple edges. For instance, introgression is an example of convergence, where two taxa (eg. species) become more genetically similar with time. This convergence cannot be modelled on regular phylogenetic trees that allow only for divergence.

Chapter 1 introduces phylogenetics, beginning with an explanation of what phylogenetics is and its importance in biology. This section will be followed by brief introductory sections to the chapters to follow. Finally, we will conclude with a review of the literature in some of the most important current topics in phylogenetics.

Chapter 2 discusses the importance of Abelian group-based models in phylogenetics and their relationships to group theory and representation theory. We will introduce the reader to Markov chains and their application to transition matrices in phylogenetics. We will then discuss the resulting properties of the transition matrices and their corresponding rate matrices and argue the importance of Abelian group-based models. We will describe how the Abelian group-based models can be diagonalised and why diagonalisation is a valuable tool for finding expressions for probability distributions in phylogenetics. We will expand the work by Hendy and Penny [1989] and Hendy [1989] on Hadamard conjugation to include a number of Abelian group-based models, some that have had their diagonalisation matrices described already by Sumner et al. [2014] and others that have not. We conclude the chapter with a collation of the diagonalising matrices for the Abelian group-based model examples.

Chapter 3 focusses on phylogenetic tensors, tensor representations of probability distributions. We show how the splitting operator, introduced by Bashford et al. [2004] and used to split an edge

on a network into two, can be pushed back to the root of the network, with all bifurcations then occurring at the root [Sumner et al., 2012c]. We will show that this process results in correlated changes, where multiple identical edges are forced to remain identical. We then argue that this result can be used to not only maintain identical edges, but also to model the convergence of diverged edges. We will call these networks *convergence-divergence networks*. This is a fairly new and unique method of modelling networks in phylogenetics. Finally, we will develop some new explicit expressions, which will be represented as polynomials expressions in Chapter 4 and Chapter 6, for the phylogenetic tensor elements of trees and networks on the binary symmetry model. These expressions will be represented in the transformed, or Hadamard, basis. This polynomial representation of the transformed phylogenetic tensor elements will be necessary for the comparison of trees and networks in Chapter 4 and Chapter 6.

Chapter 4 will be a detailed analysis of two-taxon and three-taxon trees and convergence-divergence networks. We will address the issue of identifiability of models and introduce some new terminology. We will raise the question of whether different network “structures” can be *distinguished* from one another. For example, if the set of all pattern frequencies that could have arisen on one tree or network is identical to the set of all pattern frequencies that could have arisen on the other tree or network then we say that they are *indistinguishable*. If the sets of all pattern frequencies are not identical then we say that the two trees or networks are *distinguishable*.

Chapter 5 will examine the permuting of taxon labels on the three-taxon convergence-divergence network with convergence between “non-sister” taxa, taxa that do not share a cluster other than the cluster containing all taxa. We will look at whether appropriate time parameters can still be found on the network if we permute the taxon labels. We will compare maximum likelihoods for the taxon label permutations on the network.

Chapter 6 will discuss methods of extending the work on distinguishability in Chapter 4 to higher taxon trees and networks. We will discuss how concepts from algebraic geometry, namely ideals and Gröbner bases, can be utilised to extend our work in the two-taxon and three-taxon cases to trees and networks involving a higher number of taxa. We will refer back to our three-taxon examples to illustrate the use of Gröbner bases, before arguing that Gröbner bases provide a more feasible way of analysing more complicated, higher taxon networks. Finally, we will examine a number of four-taxon trees and networks and determine whether these trees and networks are distinguishable from each other. We will conclude the chapter by showing that our algebraic geometric techniques can be applied to a very complicated convergence-divergence network with multiple convergence periods.

In the final chapter, Chapter 7, we will discuss the major findings from the previous chapters and outline the future work that could follow on from the work done in this thesis.



### 1.1.2 Phylogenetics

Phylogenetics is the study and categorisation of the evolutionary history of groups of organisms. These groups, called taxa, are genetically distinct from one another, such as different species, genera or subspecies. Taxa are classified according to a hierarchy. The hierarchy reflects the assumption that distinct taxa evolved from a common ancestor in the hierarchy and have shared traits. With progression from the time of the common ancestor, each taxon gradually acquired different traits from each other taxon, eventually resulting in the diverse range of life on Earth today.

Most commonly, a rooted tree is used to display the evolutionary history of a number of taxa. Rooted trees start with a common ancestor of all taxa on the tree, called the root. Edges coming from the root represent groups of taxa which have evolved independently of, or diverged from, each other after splitting apart at the root. Further splits of edges can occur further down the tree at positions called nodes, with further divergence of edges below these nodes. The bottom of the tree represents the taxa present at the final stage in the tree, the leaves. This is often taken to be the present time, but more generally can be any cross-section in time. Times between nodes, including the root, the leaves of the tree and all internal nodes, are often included. A common task in phylogenetics is to find the “structure” of a phylogenetic tree and the lengths of its edges, representing the divergence times. In Figure 1.1 below is a simple example of a phylogenetic tree.

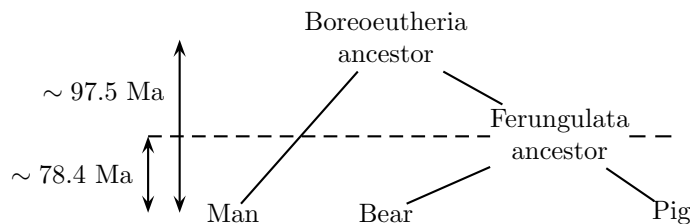


Figure 1.1: A simple phylogenetic tree representing the genetic relationships between three species: man (*Homo sapiens*), bear (*Ursus arctos*) and pig (*Sus scrofa*), using divergence time estimates from Hedges et al. [2015].

Time progresses down the page from the ancestral Boreoeutheria species at the root to the three present day species: man, bear and pig. The most recent common ancestor of the Boreoeutheria clade, which includes man, bears and pigs, lived approximately 97.5 million years ago, while the most recent common ancestor of the Ferungulata clade, which includes bears and pigs, lived approximately 78.4 million years ago. It is usual for the horizontal axis to hold no meaning. However, in this thesis we will allow the horizontal axis to represent how much divergence from a common ancestor has occurred and how much subsequent convergence has occurred. The closer two taxa are on the horizontal axis the less they have diverged from a common ancestor or the more convergence has occurred. Swapping the order of two edges below a node has no consequence. For example, we could have swapped the labels for bear and pig and the tree would

have the same meaning.

Sumner et al. [2012c] argued that convergence-divergence networks could be used to model convergence. In this thesis, our convergence-divergence networks can be thought of as regular phylogenetic trees with some added convergence periods. When two or more taxa are no longer diverging from each other, but are becoming more similar over time, we say they are converging. Examples of biological processes that convergence-divergence networks could be used to model are introgression and horizontal gene transfer. Alternatively, convergence-divergence networks could be used to model convergence of morphological characteristics.

Anderson and Hubricht [1938] and Anderson et al. [1949] define introgression as the exchange of alleles from one genome to another through hybridisation or backcrossing. Introgression may result in a more complicated mixture of genomes than a single occurrence of hybridisation between two organisms.

Keeling and Palmer [2008] describe horizontal gene transfer as the exchange of genetic material between genomes not related to reproduction. Genetic material from one genome can be added to another or exchanged between the genomes. Horizontal gene transfer is well known in bacteria, however it also occurs in eukaryotes. Andersson et al. [2003] argue that mitochondria, found in most eukaryotes, came from alpha-proteobacteria. Furthermore, they argue that despite the mitochondria and nuclear genomes being distinct, horizontal gene transfer has occurred between them.

Analogous morphological characteristics can arise in taxa that are not related due to natural selection [Parker et al., 2013]. An example of convergence of analogous morphological traits is wings in bats and birds. Theißen [2002] argues that although bats and birds inherited forelimbs from a tetrapod common ancestor, they are not closely related and have evolved wings for flight separately due to similar selective pressures.

When attempting to make comparisons between different types of organisms we must first separate these organisms into different taxa, with each taxon being a genetically distinct group of organisms. An example of taxon separations is the commonly used taxonomic hierarchy, which places species at the bottom and domain (*Eukaryota*, *Bacteria* and *Archaea*) below life at the top. An example of taxonomic distinctions at the species level is the *Macropus* genus. This genus includes many different Kangaroo *species* and also contains wallaroo species and some wallaby species [Wilson and Reeder, 2005]. Taxa can be distinguished further beyond the taxonomic hierarchy, for example at the subspecies level or at the subgenus level between genus and species.

In recent decades, organisms have often been assigned to a particular taxon based on genetic evidence. Sometimes, however, distinctions are made from other types of evidence, such as morphological differences found in fossils for which the DNA has degraded over time. An example of morphological differences is the presence or absence of vertebrae in animals. The *Vertebrata* subphylum is an example of equivalent genetic and morphological taxonomic distinctions.

A question that often arises in molecular phylogenetics is where a set of taxa fit on a phylogenetic tree or network. To answer this question, we usually start with a character sequence

for each taxon. A character sequence is a string of characters, with all characters being the same type of object. The state that each character takes must be from the same state space. For example, the state space for nucleic acid sequences is the four DNA nucleobases described earlier, C, G, A and T. The four nucleobases are the same type of object. We cannot include, for example, a state for the presence or absence of vertebrae into this state space.

### 1.1.3 Markov Models

Markov models are often used in phylogenetics to model the rates of changes between states in the state space. If these changes in state obey the Markov property then we can use a Markov model. For example, if the state space is the four nucleobases, C, G, A and T, then the substitution rates between the nucleobases can be modelled as a Markov chain.

A random walk is a process in which some object, which takes a state, changes its state randomly. These state changes can happen at fixed time intervals, known as discrete time intervals, or at random times intervals, known as continuous time intervals. In phylogenetics, the object we are dealing with may be a single character in a character sequence. For example, the state of our character may be one of the DNA nucleobases (cytosine (C), guanine (G), adenine (A) and thymine (T)). Suppose the initial state of the character at a position, or site, in the genome of a species is G. After some random period of time there may be a character substitution into the state T in the genome. At some further random time, another substitution may change the state back to G. Furthermore, after another random period of time, the character may change state to T. This is an example of a random walk in phylogenetics.

The outcome of a step in the random walk may or may not depend on the previous state that the object took. From our example, the final substitution was from state G to state T. This substitution may or may not depend on the history of the random walk, that is the previous substitutions from G to T to G. If we assume that all state changes depend only on the current state that the object takes and not on any previous states then we say that the object obeys the Markov property and we call it a Markov chain. We often assume processes in phylogenetics to be Markov in nature. If the joint probability distribution is assumed to obey a Markov process then the probabilities for transitions between states can be organised into stochastic transition matrices. For ergodic random walks, where every state in the state space can be reached from every other state in some finite number of steps,  $s$ , it is shown that after a certain number of substitutions, further substitutions depend very little on the initial state [Diaconis, 1988]. We can therefore assume that if enough substitutions have occurred then an objects' state changes display Markov behaviour.

We call the position in the sequence that a particular character is its site. Generally each site is assumed to behave independently of every other site, and with an identical process. This is called the assumption of independent and identically distributed random variables (i.i.d. assumption), a very commonly used assumption in phylogenetics. The consequence of the independence property of the i.i.d. assumption is that the behaviour of each site can be modelled by a Markov chain,

which is independent of every other site. The consequence of the identical distributions property is that the probability of a substitution from one state in the state space to another state in the state space over a given period of time must be equal for every site in the sequence and independent of every other site in the sequence. What it does not mean is that every character must be the same at each site in the sequence. In Table 1.1 below is an example of three aligned nucleic acid sequences, each with ten sites.

C	<b>T</b>	<b>C</b>	G	<b>T</b>	A	G	T	G	C
C	A	G	G	<b>G</b>	A	G	<b>A</b>	G	C
C	A	G	G	<b>C</b>	<b>T</b>	G	T	G	C

Table 1.1: An example of three aligned nucleic acid sequences, each with ten sites. Each site is designated by its own column. Sites which differ between sequences are highlighted in boldface.

We can now use a Markov model to describe the behaviour of the entire set of taxa. Markov models are square stochastic matrices, with elements representing either the probabilities or rates over time of substitutions between states in the state space. If the rate matrices can be represented in terms of the elements of an Abelian group then the model is *Abelian group-based* or simply *group-based*. Abelian group-based models have some desirable mathematical properties that will be discussed in more detail in Chapter 2. Group-based models can be diagonalised by a diagonalising matrix. When the diagonalising matrix is a Hadamard matrix, first described in a form by Sylvester [1867] and later expanded upon by Hadamard [1893], we refer to the diagonalisation process as *Hadamard conjugation* or a *Hadamard transformation*. Hadamard conjugation was first introduced by Hendy and Penny [1989] and Hendy [1989]. Exponentiation of the resulting diagonalised rate matrix is then straightforward. We will explain in Chapter 2 the desirable effects that diagonalisation has on the probability distribution and finding the diagonalising matrices for a number of Abelian group-based models.

The simplest Markov model of evolution for sequences of four states was described by Jukes and Cantor [1969] and assumes the rates of substitutions to be equal for every substitution. In other words, the rate matrix representing the substitution rates has only one parameter. If we assume some initial probability distribution of character states at the root of the tree or network, the earliest position in time, then we can use a Markov model to model substitutions between character states leading from the root to a node, where one taxon splits into multiple, usually two, descendant taxa. The Markov model must assume constant rates of substitutions between characters from the root to the node. We can model every edge between two nodes in this fashion, with nodes including the root and leaves of the tree or network. If two sequences are known, each for a different taxon, with no other information available, it cannot be determined if the sequences are ancestral and descendant to each other nor which sequence is ancestral and which is descendant. For this reason time-reversible Markov models are often preferred when the root position is unknown, as the root of the tree can be placed in an arbitrary position. Almost

always, the model parameters are fixed across a tree or network. When all parameters are equal on the tree or network we say that the process is homogeneous. This raises the question of what happens when a Markov model on an edge is partitioned into several successive Markov models comprising the length of the edge. Sumner et al. [2012a] showed that for some Markov models the combination of many successive Markov processes along the edge cannot be modelled by a single process along the edge under that same model. The general time-reversible model was shown to be one of these models lacking closure.

If we have a set of aligned sequences, each representing a different taxon, we can compare each taxon by comparing the states that each sequence takes at each site independently. The more sites taking the same state between two sequences, the closer those two taxa will be on a phylogenetic tree or network. The limit is where two sequences are identical, as would be the case with identical twins, assuming no new substitutions have occurred in their DNA. The fewer sites taking the same state between two sequences, the further apart those two taxa will be. With a set of more than two taxa we can determine where each taxon should be placed on the tree or network.

#### 1.1.4 Phylogenetic Tensors on Convergence-Divergence Networks

Suppose we are given a phylogenetic tree or network. We want to know what the theoretical probability distribution representing the tree or network in the diagonalised, or transformed, basis will be. In Chapter 3 we will discuss the relevant algebra and notation needed to express the theoretical probability distribution of a given tree or network. We call the probability distribution tensor representing the probability distribution the phylogenetic tensor. If there are  $m$  states in the state space and each of the  $n$  taxa can take any of these  $m$  states at a particular site, then there will be  $m^n$  elements in the phylogenetic tensor.

To derive explicit expressions for the phylogenetic tensor we start with the initial probability distribution, the probabilities of each state at the root, where there is just one taxon, the taxon ancestral to all other taxa. In practice, the initial probability distribution is rarely known and the probability distribution is only known at the leaves, the most recent time. We must therefore make a choice for our initial probability distribution. The choice made is often the stationary distribution. The stationary distribution represents the long run probability distribution for each individual taxon.

The next task after deciding on the initial probability distribution is to model the splitting of the single root taxon into the multiple descendant edges. In this thesis we will only be examining bifurcating trees and networks, where all nodes, including the root, split into two descendant edges. Bashford et al. [2004] introduced the *splitting operator*, an operator which takes a single edge at the root or a node and instantaneously outputs two descendant edges.

An Abelian group-based Markov model is then utilised as described in the previous section on Markov models. The Abelian group-based Markov model is applied to every edge of the tree. From here, the phylogenetic tensor for any tree formed by an Abelian group-based Markov model

can be found. Sumner et al. [2012c] showed that the splitting operator can be used on an Abelian group-based Markov model to model convergence, allowing us to find the phylogenetic tensor for networks allowing convergence. We will call these networks *convergence-divergence networks*.

Convergence-divergence networks are not equivalent to any previous methods of generalising the structure of trees. Unlike methods based on splits it is directed in time. It is also different from the approaches for implementing maximum likelihood on networks that are described in Nakhleh [2011]’s review. In Nakhleh [2011]’s review a network is thought of as encoding a set of trees (those displayed by the network). The likelihood is then either a mixture model over these trees [Jin et al., 2006], or each site is allowed to pick the tree that suits it best. Convergence-divergence networks will clearly have different limiting properties to either of these implementations. If the convergence process is run for a long enough period of time then the taxa that are converging can get arbitrarily close - this is not the case in either the mixture-model setting or for the  $n$ -taxon process as described by Bryant [2009].

The convergence-divergence models provide a lot of flexibility, perhaps too much flexibility. Before suggesting that they are a practical tool for phylogenetics we need to address two points that were well put by Steel [2005]. Steel’s two key points to keep in mind when developing models are:

1. Are they capturing a process that is important biologically?
2. Do they overfit the data?

As soon as we leave the safe waters of tree-based inference where the number of “structural” parameters we need is determined by the number of species under consideration (e.g. in a rooted binary tree with  $n$  taxa, we know we need  $2n - 2$  edges) we have to think much more carefully about variable selection.

Addressing the first point, it seems that the convergence-divergence model might be appropriate for modelling introgression, in the extreme case leading to despeciation. In this sense, the model can be thought of as a species-level analogue to the population-level isolation/migration model of Hey [2010]. Seehausen et al. [2008] argued that a loss of diversity can break down ecological boundaries, allowing more opportunities for the exchange of genetic material among previously independent populations, which can in turn lead to convergence. Taylor et al. [2006] described a case where environmental changes may be resulting in the convergence of three-spined sticklebacks (*G. aculeatus*) in Enos Lake, Vancouver Island. Sheppard et al. [2008] and Sheppard et al. [2011] identified a case in which two species of bacteria, *C. jejuni* and *C. coli*, appear to be in the process of undergoing convergence through horizontal gene transfer. A further scenario where we might consider applying convergence-divergence networks is for morphological data where selection acts similarly on taxa on different parts of the tree, causing some of the morphological characters to converge.

We will address the second point from Steel [2005] in this thesis. Can we put limits on what structures are sensible to consider with these models? We examine some cases on two, three and

four taxa and determine in what circumstances parameters are recoverable. In other words, given some particular time periods, is there a one-to-one map between the time parameters and the probability distribution on the pattern frequencies? This is a question of *identifiability*. Given a two-taxon network where taxon 1 and taxon 2 diverge for time  $t_1$  and then converge for time  $t_2$ , is there an equivalent time parameter where they diverge for time  $t'$ ? This is a question of *network identifiability*. Given a pattern frequency, could that pattern frequency have arisen on either of the trees or networks? If the set of pattern frequencies that could have arisen on one tree or network is identical to the set of pattern frequencies that could have arisen on the other tree or network then we say that the two trees and networks are *indistinguishable*. If the sets of pattern frequencies are not identical then we say that the two trees and networks are *distinguishable*. Generally if two trees or networks are not distinguishable we will choose the tree or network which is least parameter rich or is most biologically feasible.

A framework will be developed to find the phylogenetic tensors and the individual phylogenetic tensor elements for a given Abelian group-based Markov model and a given tree or network. We will show that we can use the algebraic properties of a Markov model to find explicit expressions for the phylogenetic tensor in the transformed basis. Simple substitutions for the exponential terms can turn the phylogenetic tensor elements into non-linear polynomial equations. In the transformed basis some of the elements of the phylogenetic tensor will always be constant. These constraints are similar to phylogenetic invariants, introduced by Cavender and Felsenstein [1987] and Lake [1987]. We will also show that some of the constraints are variable, similar to the Fourier transform inequalities described by Matsen [2009].

Arbitrarily many convergence-divergence networks could be constructed under the framework we will describe. For example, we could have a network where a process involving divergence of a group of taxa followed by convergence of the same group is then repeated arbitrarily many times. We will put a limit on convergence and divergence periods and restrictions on when and how they can occur.

In Chapter 4 and Chapter 6 we will examine various two-taxon, three-taxon and four-taxon trees and networks and determine which are distinguishable from each other. We will determine the sets of equality and inequality constraints on the transformed phylogenetic tensor for each tree and network. We will determine whether the trees and networks are distinguishable from each other by comparing their sets of constraints. If the two sets of constraints are identical then the two trees or networks are indistinguishable. If the two sets of constraints are different then the two trees or networks are distinguishable. In the case that the two trees and networks are distinguishable we will determine the intersection between these two sets of constraints.

As well as comparing different trees and networks, we will compare the time parameters on different taxon label permutations on the convergence-divergence network with convergence between non-sister taxa in Chapter 5. Given a set of time parameters on one taxon label permutation we will determine whether an equivalent set of time parameters can be found on another taxon label permutation that satisfies the same constraints and satisfies the condition that time

parameters must be non-negative. Finally, we will show how the maximum likelihood on one taxon label permutation can be compared to the maximum likelihood on another taxon label permutation.

### 1.1.5 Ideals and Gröbner Bases

In Chapter 4 we determine the distinguishability of many two-taxon and three-taxon convergence-divergence trees and networks on the binary symmetric model. These examples are simple enough that we can compare the phylogenetic tensor elements of each tree and network using elementary algebra. By making simple substitutions, we can turn exponential equations into non-linear polynomial equations, as shown in earlier chapters, which are relatively straightforward to solve.

For more complicated Abelian group-based models than the binary symmetric model or for trees and networks with more than three taxa, it will quickly become infeasible to continue using elementary algebraic methods. In Chapter 6 we will discuss a more general method for dealing with our phylogenetic tensor equations than was used in Chapter 4. As was the case with the three-taxon trees and convergence-divergence networks, all phylogenetic tensor elements for four-taxon trees and networks will be able to be expressed as non-linear polynomial equations by making simple substitutions to convert the exponential expressions into polynomial expressions. Once we have the polynomial equations, we can rearrange them to be identically zero and express the polynomials from the non-zero sides of these equations as generators on an ideal. We can then find a different set of generators that generates the same ideal. The polynomials from this set of generators will have the same solutions as the polynomials from the original set of generators when expressed as polynomial equations identical to zero. We will show that if the basis of the ideal is the *Gröbner basis* then we can solve the polynomial equations from the polynomial generators and find all of the inequality and equality constraints on the tree or network. We will compare various four-taxon trees and networks and determine which trees and networks are distinguishable from each other.

## 1.2 Literature Review

### 1.2.1 Networks in Phylogenetics

Networks play a role in phylogenetics as an alternative to or extension of phylogenetic trees. A network may be appropriate when the best representation of the evolutionary history of a set of taxa is multiple trees or a generalisation of a single tree to no longer require independent evolution of all edges. Networks may be desirable when convergent evolution, or despeciation, has occurred. Introgression, horizontal gene transfer and recombination can all lead to convergent evolution. Networks can also be used to represent model heterogeneity, sampling error and uncertainty caused by parallel evolution, where unrelated taxa evolve similar traits independently through similar ecological pressures. A network may be preferred even when a single tree may be the best



model for the evolutionary history of a set of taxa. If this tree is unknown and several candidate trees are roughly equally good fits, then this information can be represented in a network.

Much of the work on networks in phylogenetics up until now has dealt with *splits*, where taxa are bipartitioned into non-intersecting sets based on their evolutionary history. The structure of a tree can be represented by a series of splits, with each split representing an edge on the tree. However, these splits are not necessarily required to fit on a tree, but must fit on a more general *split network*. These split networks may not necessarily have splits which are all *compatible* with each other. In other words, a taxon may appear on different sides of two different splits, grouped with one set of taxa on one split and away from the same set of taxa on the other split. In a tree, a taxon must group with a set of taxa on every split or away from that set of taxa on every split.

Bandelt and Dress [1992] introduced the split decomposition method. The split decomposition method takes a set of distances and returns a weakly compatible split system. It is a generalisation of the four-point condition. Suppose there are  $n$  taxa, with distances,  $d(a, b)$ , between each pair of taxa,  $(a, b)$ . For every set of four taxa,  $i, j, k$  and  $l$ , the condition,  $d(i, j) + d(k, l) < \max(d(i, k) + d(j, l), d(i, l) + d(j, k))$ , must be met.

Posada and Crandall [2001] provided a summary of many of the network methods available in phylogenetics. A thorough and comprehensive textbook for networks in phylogenetics is Huson et al. [2010]. Reticulograms [Legendre and Makarenkov, 2002] and ancestral recombination graphs [Song and Hein, 2004] are some examples of network approaches. Legendre and Makarenkov [2002] heuristically created a reticulation network from an underlying tree, adding reticulations if they improved the fit. Baroni et al. [2005] used a directed graph (digraph) approach, called “hybrid phylogenies”. Lapointe [2000] developed graphs and used four methods: pyramids, weak hierarchies, split graphs and reticulograms. Hein [1990, 1993], Wang et al. [2001], Gusfield et al. [2003, 2004], Nakhleh et al. [2004] explored how to recreate a reticulate network if it is assumed that a set of taxa evolved under a reticulate network. Hudson and Kaplan [1985], Myers and Griffiths [2003], Bafna and Bansal [2004], Gusfield et al. [2004] looked at finding bounds on the number of reticulations needed to fit a set of data.

Maddison [1997] used a “separate analysis” technique. A reticulation network with a single reticulation corresponds to two gene trees inside the network. Each gene evolves according to one of the gene trees and the two gene trees differ by one rooted subtree prune and regraft (rSPR) step. If both gene trees are given, a reticulation network can be found for them. If there are multiple reticulations, then the number of rSPR steps connecting the two gene trees is no greater than the number of reticulations.

Nakhleh et al. [2004] developed two polynomial time algorithms from the work of Maddison [1997]. The first algorithm takes two gene trees inside a network and outputs a reticulation network. Although the algorithm allows for any number of reticulations, cycles in the reticulation network must be “galled” (node-disjoint). The second algorithm computes a reticulation network with a single reticulation and can be used when there are errors in the inferred gene trees.

Huson et al. [2005] provided an algorithm for computing a most parsimonious reticulate network. The algorithm depends on “tangles”, reticulation cycles that have at least one edge in common. The algorithm takes polynomial time if the reticulations in any tangle obey an overlapping property and if the maximum number of reticulations in a tangle is a constant. They also provided an algorithm for determining the structure of reticulate networks and a statistical procedure to determine whether reticulations occurred due to hybridisation or other means, for example lineage sorting or tree estimation error.

### 1.2.1.1 Split Networks

The following section on split networks follows the notation used by Huson et al. [2010].

Every edge on an unrooted tree can be interpreted as a bipartition of the set of all taxa on the tree. As every taxon is represented exactly once on the tree, each taxon must be represented on exactly one side of each edge. This idea of separating taxa based on edges gives rise to the concept of *splits*. We will see that splits are not only useful on unrooted trees, but can be applied to more general *split networks*.

**Definition 1.2.1 (Split).** *A set of taxa,  $\mathbb{N}$ , can be bipartitioned into a **split** of two sets,  $A$  and  $B$ , written as  $A|B$ , which are both non-empty, do not intersect and have the set of all taxa as their union,*

$$\begin{cases} A, B & \neq \emptyset, \\ A \cap B & = \emptyset, \\ A \cup B & = \mathbb{N}. \end{cases}$$

The two sets of the split,  $A$  and  $B$ , are called the *split parts*. The ordering of a split is inconsequential. Each taxon can always be separated from all other taxa in what is called a *trivial split*. If a set of splits can be displayed as an unrooted tree, the set of splits are said to be *compatible* with each other. To read more about splits see Definition 5.2.1 in Huson et al. [2010].

**Definition 1.2.2 (Compatibility).** *Suppose we have the two splits,  $A|B$  and  $C|D$ , where  $A$ ,  $B$ ,  $C$  and  $D$  are sets of taxa. The two splits are **compatible** if one of the intersections of the split parts,  $A \cap C$ ,  $A \cap D$ ,  $B \cap C$  or  $B \cap D$ , is the empty set. If none of the intersections are the empty set then the two splits are incompatible. If every pair of splits in a group of splits is compatible then the group of splits is compatible.*

If we have a set of compatible splits for every possible combination of the set of taxa, these splits can always be represented on an unrooted tree. To read more about compatibility see Definition 5.3.1 in Huson et al. [2010].

**Theorem 1.2.3 (Compatibility Theorem).** *For an  $n$ -taxon unrooted binary tree, there will be  $2n - 3$  edges, of which  $n$  will be edges representing taxa at the tips and  $n - 3$  will be internal*

edges. A unique  $n$ -taxon unrooted binary tree will exist if and only if a compatible split system exists, with  $2n - 3$  splits, of which  $n$  will be trivial splits corresponding to the tips of the tree and  $n - 3$  will be non-trivial splits corresponding to the internal edges of the tree.

*Proof.* Proof can be found in Buneman [1971]. □

From this point onwards it can be assumed that all trees we are dealing with are binary, unless specifically stated otherwise. From a combinatorial point of view it is sufficient to only consider binary trees since multifurcating trees can be constructed from binary trees by setting the appropriate edge lengths to zero.

A set of all trivial splits is a set of compatible splits, however unless  $n = 3$  it is not enough to define an unrooted tree. In general, as a consequence of the split equivalence theorem, a set of compatible splits must define at least one unrooted tree and possibly multiple unrooted trees. For example, suppose we have a four-taxon unrooted tree. There are four trivial splits, each corresponding to one taxon. However, one more compatible split, out of three possibilities, is required to define an unrooted tree. To read more about the compatibility theorem see Theorem 5.3.2 in Huson et al. [2010].

A similar concept to compatibility is *weak compatibility*.

**Definition 1.2.4 (Weakly Compatible).** Three sets of splits,  $A|B$ ,  $C|D$  and  $E|F$ , where  $A$ ,  $B$ ,  $C$ ,  $D$ ,  $E$  and  $F$  are sets of taxa, are **weakly compatible** if at least one of the intersections of the split parts,  $A \cap C \cap E$ ,  $A \cap D \cap F$ ,  $B \cap C \cap F$ , or  $B \cap D \cap E$ , is empty and at least one of the intersections of the split parts,  $B \cap D \cap F$ ,  $B \cap C \cap E$ ,  $A \cap D \cap E$ , or  $A \cap C \cap F$ , is also empty.

Weakly compatible splits can be formed from algorithms that attempt to create sets of incompatible splits from data sets that do not allow for compatible split systems. To read more about weakly compatible splits see Definition 5.8.1 in Huson et al. [2010].

We have so far only discussed unrooted trees. While splits only apply to unrooted trees, a similar concept, called *clusters*, corresponds to rooted trees. Clusters define groups of similar taxa, while splits separate taxa according to their differences. We can transform a compatible set of splits into a compatible set of clusters if we wish to generate a rooted tree. Firstly, the root is placed on an arbitrary position on the tree. Every set of taxa below a node, including the root, is then assigned to a cluster. For example, if there is a bifurcation below a node then all of the taxa below the node will form a cluster. All of the taxa below the root, that is all of the taxa on the rooted tree, will also form a cluster. We will look at the four-taxon unrooted tree in Figure 1.2 below as an example.

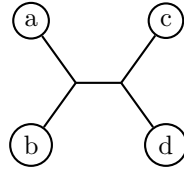


Figure 1.2: A four-taxon unrooted tree.

The splits for this tree are

$$\left\{ \begin{array}{l} \{a\} \mid \{b, c, d\}, \\ \{b\} \mid \{a, c, d\}, \\ \{c\} \mid \{a, b, d\}, \\ \{d\} \mid \{a, b, c\}, \\ \{a, b\} \mid \{c, d\}. \end{array} \right. \quad (1.1)$$

Suppose we now root the tree by placing a root on an arbitrary edge. Let's put the root on the edge representing the split  $\{a, b\} \mid \{c, d\}$  and label it  $r$ . The rooted tree is displayed in Figure 1.3 below.

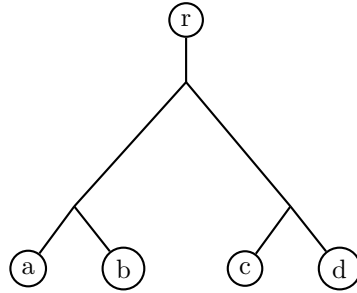


Figure 1.3: A four-taxon rooted tree.

The clusters for this tree are the sets of taxa under each node, including the leaves,

$$\left\{ \begin{array}{l} a, b, c, d, \\ \quad a, b, \\ \quad \quad c, d, \\ \quad \quad \quad a, \\ \quad \quad \quad b, \\ \quad \quad \quad c, \\ \quad \quad \quad d. \end{array} \right. \quad (1.2)$$

For the inverse problem, we wish to go from a set of clusters to a set of splits. Supposing the set of clusters is compatible, they must correspond to a rooted tree. The set of splits can be found by simply unrooting the tree and finding the set of splits by dividing the unrooted tree along every edge. With incompatible clusters the task of finding a set of splits becomes more challenging. Suppose we have three taxa:  $a$ ,  $b$  and  $c$ . Suppose we have the two clusters:  $a, b$  and  $b, c$ . These two clusters are incompatible since they cannot be represented on the same rooted tree. The corresponding set of splits, however, is compatible since they can all be represented on the same unrooted tree, shown below in Figure 1.4.

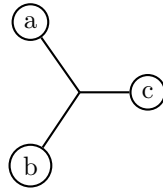


Figure 1.4: A three-taxon unrooted tree.

To turn the set of clusters into a set of splits we first introduce a dummy taxon set,  $r$ . The dummy taxon set is placed with the split part that does not contain the cluster. In our example, the splits become  $\{a, b\} | \{c, r\}$  and  $\{b, c\} | \{a, r\}$ , which can be represented on the two incompatible unrooted trees shown in Figure 1.5 below, each corresponding to one of the splits.



(a) A four-taxon unrooted tree representing the non-trivial split,  $\{a, b\} | \{c, r\}$ .  
 (b) A four-taxon unrooted tree representing the non-trivial split,  $\{b, c\} | \{a, r\}$ .

Figure 1.5: Two incompatible four-taxon unrooted trees, each corresponding to a different split.

If a set of splits are incompatible, such as the two trees above, and we wish to obtain a tree, some of the splits or some of the taxa can be removed until the splits are compatible. Alternatively, a *split network* may be preferred instead.

A split network can be derived from a *split graph*, a finite, connected graph. To get from a split graph to a split network we apply taxon labels to the leaves and split labels to the edges. Single edges represent splits which are compatible with every split on the graph. Sets of parallel edges represent splits which are incompatible with at least one split on the graph. Removing every edge corresponding to a particular split will result in two new split graphs, with each split graph representing one split part.

If the split network corresponds to an unrooted tree then every split will be represented by a single edge and removing one of these edges will result in the two new split graphs, both corresponding to unrooted trees. Each unrooted tree will contain the set of taxa from one partition of the split, with the two unrooted trees not sharing any taxa.

Split networks are not unique. There may be multiple split networks all representing the same set of splits. A split network for a compatible set of splits may not be an unrooted tree, however an unrooted tree will always exist for a compatible set of splits.

To see how incompatible splits on split networks work, let's look at an example. Suppose we have the six-taxon split graph displayed in Figure 1.6 below.

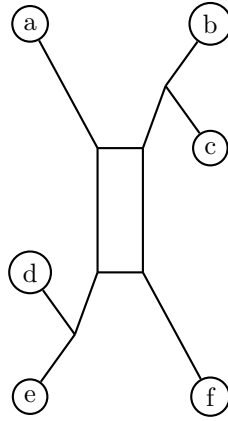
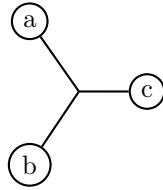
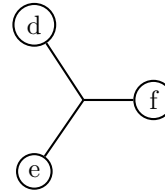


Figure 1.6: A six-taxon split graph.

There are two sets of parallel edges, each representing a split which is incompatible with at least one other split on the graph. The two parallel vertical edges represent the split,  $\{a, b, c\} | \{d, e, f\}$ . The two parallel horizontal edges represent the split,  $\{a, d, e\} | \{b, c, f\}$ . Suppose we now removed the two edges corresponding to the split,  $\{a, b, c\} | \{d, e, f\}$ . We now have two split graphs, each representing one of the split parts, shown in Figure 1.7 below.



(a) The remaining three-taxon split graph representing the split part  $\{a, b, c\}$  after the edges representing the split  $\{a, b, c\} | \{d, e, f\}$  are removed.



(b) The remaining three-taxon split graph representing the split part  $\{d, e, f\}$  after the edges representing the split  $\{a, b, c\} | \{d, e, f\}$  are removed.

Figure 1.7: Two split graphs, each representing one of the split parts.

We could also remove the two edges representing the split  $\{a, d, e\} | \{b, c, f\}$ , shown in Figure 1.8 below.

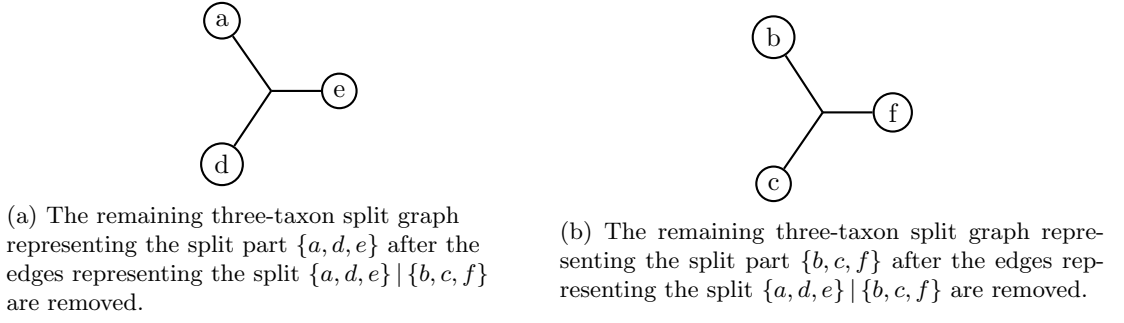


Figure 1.8: Two split graphs, each representing one of the split parts.

To see why the two splits are incompatible with each other, we need to compare the split parts from the two splits. The intersections of the split parts are

$$\begin{cases} \{a, b, c\} \cap \{a, d, e\} &= \{a\} \neq \emptyset, \\ \{a, b, c\} \cap \{b, c, f\} &= \{b, c\} \neq \emptyset, \\ \{d, e, f\} \cap \{a, d, e\} &= \{d, e\} \neq \emptyset, \\ \{d, e, f\} \cap \{b, c, f\} &= \{f\} \neq \emptyset. \end{cases} \quad (1.3)$$

Since none of these intersections are the empty set, the two splits are incompatible with each other.

When dealing with a data set, we rarely have the option of describing the data with a compatible split system. We may need to resort to an algorithm which returns an incompatible split system instead. The first attempts at obtaining sets of incompatible splits from data sets were made using the split decomposition method, introduced by Bandelt and Dress [1992]. The split decomposition method uses a distance matrix and returns a set of weakly compatible weighted splits.

Split networks describe data sets, but do not necessarily have obvious biological meanings. For example, it is not clear what the biological meaning of a node connecting two incompatible splits is, such as on Figure 1.6 on page 17.

Directed acyclic graphs (DAGs) provide an alternative to split networks when a direction in time is known. Figure 1.12 on page 20 shows a rooted DAG, where two edges join at the same node.

Generally, split systems and cluster systems are used to describe unrooted trees and rooted trees, respectively. We can, however, also use cluster systems to describe our convergence-divergence models.

Let's look at the three-taxon clock-like tree in Figure 1.9 below as an example.



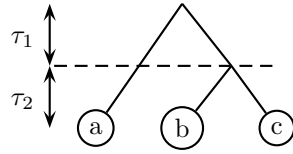


Figure 1.9: Three-taxon clock-like tree.

The clusters for this tree are

$$\left\{ \begin{array}{l} a, b, c, \\ b, c, \\ a, \\ b, \\ c. \end{array} \right. \quad (1.4)$$

Now let's look at the three-taxon convergence-divergence network in Figure 1.10 below as another example.

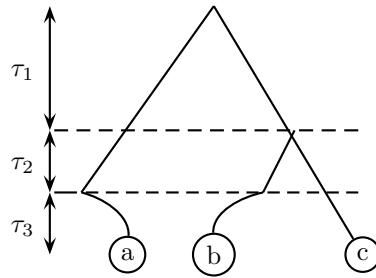


Figure 1.10: Three-taxon clock-like convergence-divergence network with a convergence period involving the first and second taxa in the third time period. Convergence is represented by curved lines.

It is not clear what the cluster system should be for this convergence-divergence network. If there is a “small” amount of convergence then the convergence-divergence network will be “similar” to the three-taxon clock-like tree that we just discussed. Therefore, taxa  $b$  and  $c$  will still form a cluster and the cluster system will remain the same. However, if a “large” amount of convergence is allowed to happen, then taxa  $a$  and  $b$  will have progressively converged towards

each other. Taxa  $a$  and  $b$  will now cluster together. The new cluster system would then be

$$\left\{ \begin{array}{l} a, b, c, \\ a, b, \\ a, \\ b, \\ c. \end{array} \right. \quad (1.5)$$

This is the same cluster system as described by the three-taxon clock-like tree in Figure 1.11 below.

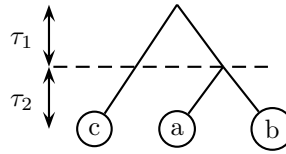


Figure 1.11: Three-taxon clock-like tree.

Our convergence-divergence networks can be interpreted as representing a particular cluster system if the “amount” of convergence is known. If it is unknown how much convergence has happened, then the convergence-divergence networks may correspond to multiple incompatible cluster systems. If sister taxa converge, then the cluster system describing the convergence-divergence network will be the same as the cluster system describing the tree with no convergence.

### 1.2.1.2 Directed Acyclic Graphs (DAGs)

Split networks generalise unrooted trees to allow for non-tree like behaviour. *Directed acyclic graphs (DAGs)* allow us to generalise rooted trees to allow for non-tree like behaviour.

**Definition 1.2.5 (Rooted DAG).** A **DAG** is a graph which is both directed, such as in time, and has no directed cycles. A **rooted DAG** contains a single root, a node which has no edges leading to it in the specified direction.

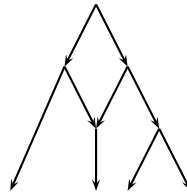


Figure 1.12: A rooted DAG. Arrows refer to the direction of time.

Rooted DAGs are described in Definition 1.4.1 in Huson et al. [2010].

The definition of a DAG does not exclude cycles, it only excludes *directed* cycles. Suppose the direction associated with the DAG is removed and we allow for any direction along an edge. There are some DAGs, such as the one above, for which a node can be returned to without the path requiring any edge being traversed twice.

Suppose we have two nodes,  $a$  and  $d$ , on a DAG. We define  $a$  as an *ancestor* of  $d$  and  $d$  as a *descendant* of  $a$  if  $d$  can be arrived at from  $a$  in the specified direction of the DAG. *Parents* and *children* are specific types of ancestors and descendants, respectively, which are separated by a single edge on the DAG. The term *lower* is used to define a descendant node in comparison to an ancestor node. We can compare edges in a similar way. If the ancestor of the first edge is lower than the descendant of the second edge then the first edge is lower than the second edge. If neither of a pair of nodes or edges is lower than the other then we call the pair of nodes or edges *incomparable*.

The root of a DAG is the only node that does not have any ancestor node. Every other node in the DAG must be a descendant of the root, with those descendant nodes which are separated by a single edge also being child nodes. Conversely, the leaves of a DAG are the only nodes that do not have any descendant nodes. However, generally only some nodes will be ancestor nodes of an individual leaf. Every internal node, that is every node that is not the root or a leaf, must have the root as an ancestor node, at least one leaf as a descendant node and potentially some other internal nodes as ancestor or descendant nodes. DAGs may also contain *sub-DAGs*. A sub-DAG fits the definition of a DAG, but is found within the DAG itself.

If the only node with no ancestor node is the root and every other node has exactly one ancestor node then the DAG is a rooted tree. If each node also has two descendant nodes, including leaves, then the tree is a rooted bifurcating tree. If we wish to obtain a rooted tree from a DAG we can selectively remove some of the edges from the DAG. For every node with multiple ancestor nodes, we must remove all but one of the edges connecting the node to an ancestor node. Let's look at Figure 1.12 as an example. There is only one node with multiple ancestor nodes. This node has two ancestor nodes. To convert the DAG into a rooted tree we must remove one of the edges connecting this node to an ancestor nodes. The two rooted trees which result are those in Figure 1.13 below.



(a) The rooted tree which remains when the right edge above the node of the DAG with two ancestor nodes is removed.

(b) The rooted tree which remains when the left edge above the node of the DAG with two ancestor nodes is removed.

Figure 1.13: The two rooted trees which remain when one of the edges is removed.

We can see that a DAG can be used in circumstances where multiple rooted trees may be of interest. Situations where this may arise are when different sections of character sequences have evolved on different trees. Clearly DAGs are preferable over split networks in situations where a root is known, a direction in time is known or pairs of ancestor and descendant nodes or edges are known. DAGs can model processes success as hybridisation, horizontal gene transfer and recombination. However, they cannot model convergence, where diverged edges no longer undergo independent evolution and instead gradually become more similar over time. To model convergence we will need our convergence-divergence model.

### 1.2.2 Identifiability

An issue that often arises in model inference in phylogenetics is the issue of inferring the appropriate tree or network and its parameters from the phylogenetic tensor. Suppose we have chosen a model of evolution and we wish to compare the phylogenetic tensors for multiple tree and network structures. Recall that the phylogenetic tensor is the probability distribution for a specified tree or network. The phylogenetic tensor is dependent on the tree or network structure, the rate parameters of the model and the edge parameters. For the binary symmetric model, for each edge parameter we are often only interested in the product of the rate and edge parameters as the two parameters cannot be resolved independently.

Allman and Rhodes [2008] defined *identifiability* with respect to both numerical parameters and models. Some other recent articles addressing the issue of identifiability from Allman and Rhodes are Allman and Rhodes [2006, 2009], Allman et al. [2008, 2009, 2011b,a, 2015]. We will discuss some of the important concepts in their articles and adapt them to our needs, in the process introducing some new definitions to help us compare tree and network structures.

**Definition 1.2.6 (Numerical Parameter Identifiability).** *Numerical parameters are **identifiable** if there is an injective map between the parameter space and the pattern frequencies.*

**Definition 1.2.7 (Model Identifiability).** *A model is **identifiable** if there is a unique tree for every possible pattern frequency from the trees that we are considering.*

Allman and Rhodes [2008] recognised that there are some issues with these definitions. The first problem they stated was that the numerical parameters may not be identifiable if the numerical parameter space is unrestricted. They used the four-taxon star tree as an example. They argued that any phylogenetic tensor arising from this tree could have arisen from any other four-taxon tree with the appropriate edge lengths. The reason for this is that any four-taxon tree, or network, can be made to be equivalent to the star tree by letting the appropriate edge lengths tend towards zero. Consequently, the space of the phylogenetic tensor for the star tree contains the spaces of the phylogenetic tensors of every other tree, as well as network.

Let's look at some four-taxon tree structures in Figure 1.14 below as an example.

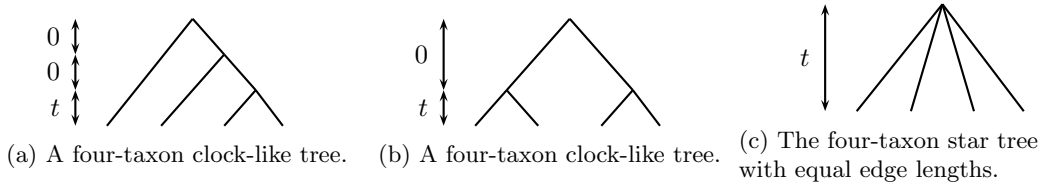


Figure 1.14: Four-taxon tree structures.

The three tree structures are clearly identical when some of the edge lengths are set to zero. Likewise, we could have identical tree structures for any other four-taxon tree, or even a convergence-divergence network, with some of the edges, including any with convergence, set to zero. We will address this issue later with a definition that we will call *distinguishability*.

We will use the definitions of Allman and Rhodes [2008], with some slight name changes. *Numerical parameter identifiability* will simply be called *identifiability*. *Model identifiability* will be called *network identifiability* to reflect our generalisation from tree structures to network structures.

Can we put limits on what structures are sensible to consider with these models? In this thesis we will examine some simple cases on two, three and four taxa and determine in what circumstances parameters are recoverable. In other words, given a particular set of time parameters, is there a one-to-one map between time parameters and the phylogenetic tensor? This is a question of **identifiability**. Now suppose we have a two-taxon network where taxon 1 and taxon 2 diverge for time  $t_1$  and then converge for time  $t_2$ . Is there an equivalent set of time parameters where they diverge for time  $t'$  with no later convergence? This is a question of **network identifiability**. Thirdly, given a set of pattern frequencies, could they have arisen on two trees or networks or only on one of the trees or networks? This is a question of **distinguishability**.

**Definition 1.2.8 (Distinguishable).** *Two networks are said to be **distinguishable** if the spaces of their phylogenetic tensors, the family of probability distributions, are not identical. That is, there are some phylogenetic tensors, or pattern frequencies, that can arise on one network, but not the other.*

To answer questions about identifiability, network identifiability or distinguishability we need

to be able to describe what the space of the phylogenetic tensors looks like. What constraints are there under different sets of time parameters? By making use of the Hadamard basis for comparisons we are able to fully describe the situation for the two and three-taxon cases, as well as some of the four-taxon cases, under the binary symmetric model.

### 1.2.3 Group-Based Models

In a series of papers from the late 80s and 90s, Hendy, Penny, Steel and co-authors showed that for Abelian group-based models, the Kimura 3 parameter model and its submodels, there was an invertible transformation between edge weights and site pattern probabilities [Hendy and Penny, 1989, 1993, Hendy, 1989, Hendy et al., 1994, Steel et al., 1992]. They called this transformation “Hadamard conjugation”. von Haeseler and Churchill [1993] were the first to note that Hadamard conjugation provided one way to think about the likelihood of more general split systems than trees. Bryant [2009] followed up on this viewpoint, showing how to extend likelihood to general split systems for the Abelian models, the so called “ $n$ -taxon process”. Bashford et al. [2004] introduced the “splitting operator”, which can be used to represent bifurcations at nodes on a tree. In their follow up paper, Sumner et al. [2012c] discussed in more detail the properties of the splitting operator, arguing that it can be applied to the general Markov model on trees and suggested that it could also be used to model convergence on a network.

A question that often arises in molecular phylogenetics is how similar sequences are and over how long those sequences have diverged from each other. With a molecular clock assumption, we can assume that the total number of substitutions in a character sequence is proportional to the time that those substitutions have occurred over. We call the total number of substitutions that have occurred from the root to all of the leaves the “evolutionary distance”. The early work of Hendy and Penny showed how Hadamard conjugation can be used to compare the evolutionary distance of a tree and the pairwise distances on a tree to the probability distribution of the tree.

To illustrate how this is done, we will use the K3ST model as an example. The K3ST model was introduced by Kimura [1981] to model three different types of nucleotide substitutions, one type of “transition” and two types of “transversions”. Transitions between the DNA nucleotides, C and T, or A and G, were all given the same time independent rate,  $\alpha$ . Transversions between the DNA nucleotides, A and T, or C and G, were given the time independent rate,  $\beta$ . The other type of transversions between the DNA nucleotides, A and C, or G and T, were given the time independent rate,  $\gamma$ . Diagrammatically, these substitutions are represented in Figure 1.15 below.

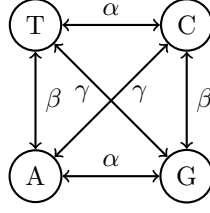


Figure 1.15: The K3ST model.

Suppose that we have two taxa that have evolved independently on a clock-like tree from a common ancestor. We wish to estimate the number of substitutions of each type that have occurred and the evolutionary distance. We cannot simply look at the two sequences and find the number of substitutions of each type. The problem with this process is that intermediate substitutions will always be ignored and the estimated number of substitutions will always be less than or equal to the actual number of substitutions. For example, suppose we can only “see” the site at the “start” and at the “end” of the process and that substitutions on a site are from C to G to A. We will only be able to see a substitution from C at the start to A at the end of the process. Assuming a Poisson process, Kimura found expressions for the expected number of each type of substitution which took into account intermediate substitutions. Hendy [2005] denoted these numbers of substitutions  $r$ , with the subscript referring to the type of substitution. The expressions they gave for the numbers of substitutions were

$$\begin{cases} r_{\emptyset} &= \frac{1}{4} [\ln(1 - 2p_{\alpha} - 2p_{\gamma}) + \ln(1 - 2p_{\beta} - 2p_{\gamma}) + \ln(1 - 2p_{\alpha} - 2p_{\beta})], \\ r_{\alpha} &= \frac{1}{4} [-\ln(1 - 2p_{\alpha} - 2p_{\gamma}) + \ln(1 - 2p_{\beta} - 2p_{\gamma}) - \ln(1 - 2p_{\alpha} - 2p_{\beta})], \\ r_{\beta} &= \frac{1}{4} [\ln(1 - 2p_{\alpha} - 2p_{\gamma}) - \ln(1 - 2p_{\beta} - 2p_{\gamma}) - \ln(1 - 2p_{\alpha} - 2p_{\beta})], \\ r_{\gamma} &= \frac{1}{4} [-\ln(1 - 2p_{\alpha} - 2p_{\gamma}) - \ln(1 - 2p_{\beta} - 2p_{\gamma}) + \ln(1 - 2p_{\alpha} - 2p_{\beta})], \end{cases} \quad (1.6)$$

where  $p$  is the time dependent probability of a substitution, with the subscript referring to the type of substitution and  $-r_{\emptyset} = r_{\alpha} + r_{\beta} + r_{\gamma}$  being the evolutionary distance.

$p_{\emptyset} = 1 - p_{\alpha} - p_{\beta} - p_{\gamma}$  is the probability that the two sequences have the same nucleotide state at any given site. The equations then become

$$\begin{cases} r_{\emptyset} &= \frac{1}{4} [\ln(p_{\emptyset} - p_{\alpha} + p_{\beta} - p_{\gamma}) + \ln(p_{\emptyset} + p_{\alpha} - p_{\beta} - p_{\gamma}) + \ln(p_{\emptyset} - p_{\alpha} - p_{\beta} + p_{\gamma})], \\ r_{\alpha} &= \frac{1}{4} [-\ln(p_{\emptyset} - p_{\alpha} + p_{\beta} - p_{\gamma}) + \ln(p_{\emptyset} + p_{\alpha} - p_{\beta} - p_{\gamma}) - \ln(p_{\emptyset} - p_{\alpha} - p_{\beta} + p_{\gamma})], \\ r_{\beta} &= \frac{1}{4} [\ln(p_{\emptyset} - p_{\alpha} + p_{\beta} - p_{\gamma}) - \ln(p_{\emptyset} + p_{\alpha} - p_{\beta} - p_{\gamma}) - \ln(p_{\emptyset} - p_{\alpha} - p_{\beta} + p_{\gamma})], \\ r_{\gamma} &= \frac{1}{4} [-\ln(p_{\emptyset} - p_{\alpha} + p_{\beta} - p_{\gamma}) - \ln(p_{\emptyset} + p_{\alpha} - p_{\beta} - p_{\gamma}) + \ln(p_{\emptyset} - p_{\alpha} - p_{\beta} + p_{\gamma})]. \end{cases} \quad (1.7)$$

It can be immediately seen that these expressions are all sums of the same three logarithmic terms, with only the signs of the terms differing between the expressions. In matrix form,

$$\begin{bmatrix} r_\emptyset \\ r_\alpha \\ r_\beta \\ r_\gamma \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ \ln(p_\emptyset - p_\alpha + p_\beta - p_\gamma) \\ \ln(p_\emptyset + p_\alpha - p_\beta - p_\gamma) \\ \ln(p_\emptyset - p_\alpha - p_\beta + p_\gamma) \end{bmatrix}. \quad (1.8)$$

Note that  $\ln(p_\emptyset + p_\alpha + p_\beta + p_\gamma) = \ln(1) = 0$ .

Recognising the Sylvester representation of the Hadamard matrix,

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}, \quad (1.9)$$

and defining the vector logarithm as the component-wise logarithm of every element of the vector, this is the equation

$$r = H_4^{-1} \cdot \ln(H_4 \cdot p), \quad (1.10)$$

where

$$r = \begin{bmatrix} r_\emptyset \\ r_\alpha \\ r_\beta \\ r_\gamma \end{bmatrix}, \quad p = \begin{bmatrix} p_\emptyset \\ p_\alpha \\ p_\beta \\ p_\gamma \end{bmatrix}. \quad (1.11)$$

Inverting the equation,

$$p = H_4^{-1} \cdot \exp(H_4 \cdot r). \quad (1.12)$$

This process is referred to as ‘‘Hadamard conjugation’’. We will see in the second chapter why the Hadamard matrix is chosen for the K3ST model. We will show that a more general matrix, for which the Hadamard matrix is contained within, will ‘‘transform’’ the K3ST model. We will expand on the Hadamard conjugation for the K3ST model and show that analogous conjugations, or transformations, can be found for any Abelian group-based model.

In this thesis we will generally be working in the transformed basis. For the K3ST model our



transformed basis will be

$$q = H_4 \cdot p = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} p_\emptyset \\ p_\alpha \\ p_\beta \\ p_\gamma \end{bmatrix} = \begin{bmatrix} p_\emptyset + p_\alpha + p_\beta + p_\gamma \\ p_\emptyset - p_\alpha + p_\beta - p_\gamma \\ p_\emptyset + p_\alpha - p_\beta - p_\gamma \\ p_\emptyset - p_\alpha - p_\beta + p_\gamma \end{bmatrix}. \quad (1.13)$$

For  $n$  taxa, we simply need the corresponding Hadamard matrix of dimension  $4^n$ . The transformed basis will be

$$q = H_4^{\otimes n} \cdot p. \quad (1.14)$$

Hadamard conjugation has the capacity to generalise model based inference beyond trees. We will later show how we can use Hadamard conjugation on our convergence-divergence models.

#### 1.2.4 Tensor Representation of Models

In Bashford et al. [2004] the splitting operator was introduced. The splitting operator represents the “branching process” and is a linear operator which takes a vector space and outputs the tensor product of the space with itself. In other words, if we have unit vectors, each representing a single character state for a single taxon, the tensor product of each unit vector with itself is another unit vector, now representing the same character on two descendant taxa. Initially after the branching process, the two taxa must be identical. Since the splitting operator is a linear operator, the action of the splitting operator on the phylogenetic tensor for one taxon is a phylogenetic tensor for the two descendant taxa, which must initially be identical.

The phylogenetic tensor for a single taxon can be represented as a sum of the probabilities over all of the possible states. For example, for a binary character state space,  $\{0, 1\}$ , the phylogenetic tensor can be represented as

$$P^{(1)}(t) = p_0(t) e_0 + p_1(t) e_1, \quad (1.15)$$

where the  $p_i(t)$  are the probabilities for the respective states after time  $t$  has elapsed from the root and the  $e_0$  are the respective unit vectors.

In Figure 1.16 below is a graphical representation of a single taxon phylogenetic tree, with time  $t$  elapsed from the root.



Figure 1.16: A single taxon phylogenetic tree, where  $\pi$  is the probability distribution at the root and  $P^{(1)}(t)$  is the phylogenetic tensor at the single leaf.

The splitting operator, denoted  $\delta$ , acts on the unit vectors as follows,

$$\left\{ \begin{array}{l} \delta \cdot e_0 = e_0 \otimes e_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = e_{00}, \\ \delta \cdot e_1 = e_1 \otimes e_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = e_{11}, \end{array} \right. \quad (1.16)$$

where  $\otimes$  is the Kronecker product.

The phylogenetic tensor for two taxa initially after the branching process is then

$$\begin{aligned} P^{(2)}(t) &= \delta \cdot (p_0(t) e_0 + p_1(t) e_1) \\ &= \delta \cdot (p_0(t) e_0) + \delta \cdot (p_1(t) e_1) \\ &= p_{00}(t) \delta \cdot e_0 + p_{11}(t) \delta \cdot e_1 \\ &= p_{00}(t) e_{00} + p_{11}(t) e_{11}, \end{aligned} \quad (1.17)$$

where  $p_{00}(t) = p_0(t)$  and  $p_{11}(t) = p_1(t)$  are scalar probability functions of time,  $t$ .

The two-taxon phylogenetic tree immediately after the branching process is shown in Figure 1.17 below.

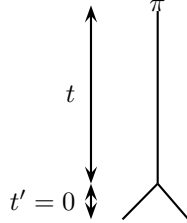


Figure 1.17: A two-taxon phylogenetic tree initially after the branching process, where  $\pi$  is the probability distribution at the root and  $t$  and  $t'$  are the time parameters.

Supposing the two edges then diverge over some time period,  $t'$ , according to a Markov model, with transition matrix,  $M(t')$ , the phylogenetic tensor then becomes

$$\begin{aligned} P^{(2)}(t+t') &= (M(t') \otimes M(t')) \cdot P^{(2)}(t) \\ &= p_{00}(t+t') e_{00} + p_{01}(t+t') e_{01} + p_{10}(t+t') e_{10} + p_{11}(t+t') e_{11}. \end{aligned} \quad (1.18)$$

Additional branching processes and diverging edges can be applied to the tree to generate any  $n$ -taxon tree that is desired. Non-clock-like trees and star trees can also be generated by applying different time periods to adjacent edges and by applying consecutive branching processes with

no divergence between them. For example, suppose we applied the splitting operator to the root of a tree. We would now have a two-taxon tree with the phylogenetic tensor being the initial probability distribution at the root on both taxa. Suppose now that we applied the splitting operator again to one of the edges descending from the root. We would now have a three-taxon tree, with all three taxa splitting from the root and with the phylogenetic tensor being the initial probability distribution at the root on all three taxa. Finally, suppose we now applied transition matrices from a Markov model to each of the three edges, but over different time periods. We would then essentially have a three-taxon non-clock-like tree or a three-taxon star tree with different edge lengths, shown in Figure 1.18 below.

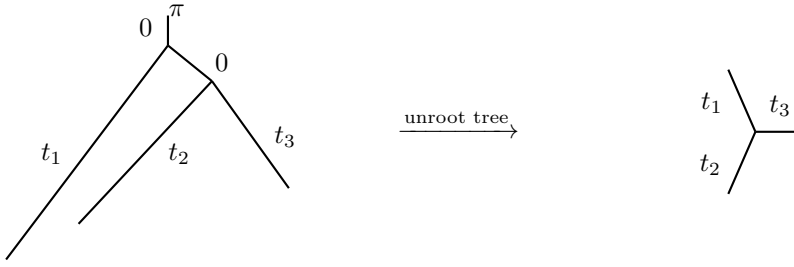


Figure 1.18: A three-taxon non-clock-like tree or star tree with uneven edges.

Recently, Sumner et al. [2012c] showed how the  $n$ -taxon process could be extended to the general Markov model. The convergence-divergence networks that they introduced can be thought of as applying combinations of two different processes to a set of  $n$  taxa. There is a convergent process which acts to make taxa more similar (or in the case of taxa that have yet to diverge it keeps them identical), and a divergent process which implements the standard idea of conditionally independently evolving edges. For models where taxa that have diverged never later converge, this is completely equivalent to the standard tree approach. We will consider what might happen if the convergent process is applied to a subset of the taxa that have previously been diverging. In its most general form the convergence-divergence model places no restriction on the number of time parameters, each with associated partitions of the taxa defining which subsets of the taxa are converging with each other and which are diverging from every other taxon.

Suppose now that we applied the action of the splitting operator to a transition matrix. We will explain the motivation for applying the splitting operator to a transition matrix later in the thesis. Using the binary symmetric model as an example, the splitting operator performs the following action on a transition matrix,

$$\delta \cdot M(t) = e^{\lambda R_{11}t} \cdot \delta, \quad (1.19)$$

where

$$\lambda R_{11} = \left[ \begin{array}{c|cccc} & 00 & 01 & 10 & 11 \\ \hline 00 & -\lambda & \lambda & \lambda & \lambda \\ 01 & 0 & -2\lambda & 0 & 0 \\ 10 & 0 & 0 & -2\lambda & 0 \\ 11 & \lambda & \lambda & \lambda & -\lambda \end{array} \right]. \quad (1.20)$$

By inspecting  $\lambda R_{11}$ , it can be seen that this rate matrix can be used to model convergence of two or more diverged edges. The rate matrix can be represented graphically, as shown below in Figure 1.19.

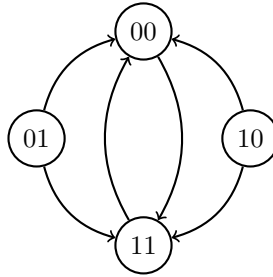


Figure 1.19:  $\lambda R_{11}$  on two taxa.

Each character state transition has the same rate,  $\lambda$ , from the binary symmetrical model.

If we want to model convergence, we must apply the convergence to two edges that have diverged away from each other. It may appear that the presence of the splitting operator immediately after the transition matrix for the convergence of two edges is problematic. Originally we had the splitting operator acting on a transition matrix, however it is no longer apparent what the splitting operator is acting on as it now appears after the transition matrix for the convergence of two edges. We call this process “pushing back” the splitting operator. We will show that every splitting operator on a tree or convergence-divergence network can be “pushed back” as many times as necessary to act on the root of the tree or network. The splitting operator is therefore not only useful for modelling bifurcations at nodes, but also the convergence of two adjacent edges. We will show later in the thesis that this convergence can be generalised to allow for convergence of any subset of  $n$  adjacent taxa.

For more detail on the algebra of the splitting operator see Bashford et al. [2004] and Sumner et al. [2012c].

## Chapter 2

# Diagonalisation of Markov Models

### 2.1 Markov Chains in Phylogenetics

Markov chains are stochastic processes acting on a state space. The defining property of a Markov chain is that the probability, or rate over time, of a transition from one state in the state space to another depends only on the current state and not on the path that was taken to arrive at that state. Markov chains are utilised in many diverse areas of mathematics involving stochastic processes, including phylogenetics. Phylogenetic methods, such as likelihood and Bayesian distance correction, rely on stochastic models of nucleotide substitutions. It is common to make some simplifying assumptions, such as the i.i.d. assumption. The i.i.d. assumption is the assumption that every character in a sequence is an independent and identically distributed random variable. Each character behaves independently of every other character and obeys the same Markov process. Each character is defined by a Markov chain, with the same Markov model of evolution, including the parameters defining the rates of mutations.

The state space is the set of all possible objects in a system. States depend on the type of characters to be considered. Examples of state spaces for biological sequences in phylogenetics are the DNA nucleobases (cytosine (C), guanine (G), adenine (A) and thymine (T)), the amino acids or a set of morphological traits. The DNA nucleobases can be partitioned into the purines (A and G) and the pyrimidines (C and T), reducing the states in the state space from four to two. Other than in this chapter, we will only examine the binary symmetric model, a state space with two states. We might choose these two states to be the purines and pyrimidines or two distinct morphological traits. Alternatively, we could use a two state model for single nucleotide polymorphisms (SNPs). SNPs are variations at a single site in nucleotide sequences within a species. Hodgkinson and Eyre-Walker [2010] state that there are usually two alleles in a SNP. Hence, a two state model is commonly used.

Mendelian traits are characterised by the presence of a dominant or recessive phenotypic trait. If an offspring receives two recessive alleles, one from each parent, on two paired chromosomes,

then they will display the recessive phenotypic trait. Alternatively, if an offspring receives at least one dominant allele then they will display the dominant phenotypic trait. The state space for Mendelian traits will contain two states, one for each allele.

In phylogenetics, mutations passed on to offspring follow a homogeneous discrete time Markov chain, since the separation between generations is a discrete period of time. However, the evolution of a population follows a homogeneous continuous time Markov chain as mutations to the genome of a population can happen continuously through time.

## 2.2 The Markov Property on Transition Matrices

The Markov property is the expression that the behaviour of a Markov chain is independent of the path taken from time  $t$  to time  $u$  and is only dependent on the states at times  $t$  and  $u$ . For any set of non-negative times,  $s_1 \leq s_2 \leq \dots \leq s_k \leq t$ , the Markov property can be expressed on a probability distribution as

$$\mathbb{P}[X(t) = i | X(s_1) = i_1, X(s_2) = i_2, \dots, X(s_k) = i_k] = \mathbb{P}[X(t) = i | X(s_k) = i_k], \quad (2.1)$$

where  $i_1, i_2, \dots, i_k, i$  are the states the Markov chain,  $X$ , takes at the respective times.

Assuming we have some process, which is not necessarily Markov, to find the probability of taking a particular final state given an initial state we must sum over all of the possible states between the initial and final states. The probability can be expressed as

$$\begin{aligned} \mathbb{P}[X(t) = i | X(s_1) = i_1] &= \sum_{i_2, i_3, \dots, i_k} \mathbb{P}[X(s_2) = i_2 | X(s_1) = i_1] \\ &\quad \mathbb{P}[X(s_3) = i_3 | X(s_1) = i_1, X(s_2) = i_2] \\ &\quad \dots \mathbb{P}[X(t) = i | X(s_1) = i_1, X(s_2) = i_2, \dots, X(s_k) = i_k]. \end{aligned} \quad (2.2)$$

In practice we may not have any knowledge of the behaviour of the process over a period of time. We may only know the initial state of the Markov chain and the final or current state. If the process is Markov then we can utilise the Markov property as follows,

$$\begin{aligned} \mathbb{P}[X(t) = i | X(s_1) = i_1] &= \sum_{i_2, i_3, \dots, i_k} \mathbb{P}[X(s_2) = i_2 | X(s_1) = i_1] \\ &\quad \mathbb{P}[X(s_3) = i_3 | X(s_2) = i_2] \dots \mathbb{P}[X(t) = i | X(s_k) = i_k]. \end{aligned} \quad (2.3)$$

We use a stochastic transition matrix, denoted  $M(t)$ , to represent the probabilities for transitions between states. The elements of the transition matrix,  $[M(t)]_{ij}$ , represent the probabilities of transitions from the states  $j$  to the states  $i$ . The notation then becomes

$$[M(s_1, t)]_{ii_1} = \sum_{i_2, i_3, \dots, i_k} [M(s_1, s_2)]_{i_2 i_1} [M(s_2, s_3)]_{i_3 i_2} \dots [M(s_k, t)]_{ii_k}. \quad (2.4)$$

In terms of the transition matrices themselves,

$$M(s_1, t) = M(s_1, s_2) M(s_2, s_3) \dots M(s_k, t). \quad (2.5)$$

It is sufficient to consider only a single intermediate time,  $u$ , where  $s \leq u \leq t$  are a set of non-negative times,

$$M(s, t) = M(s, u) M(u, t). \quad (2.6)$$

If we assume homogeneity, the transition probabilities depend only on the differences in time between the initial and final states and not on the actual times themselves. This can be expressed as

$$M(s, t) = M(0, t - s). \quad (2.7)$$

We then simplify the notation to specify only the time difference and not the initial and final times,

$$M(t) := M(0, t). \quad (2.8)$$

If no time has elapsed, the state must not change from the initial state. In other words, the probability of transitioning from the initial state to any other state must be zero and the probability of staying in the same state must be one. The transition matrix initially must therefore be the identity matrix,

$$M(0) = I. \quad (2.9)$$

After some small amount of time has progressed there will be non-negative probabilities of transitions away from the initial state. The probability of staying in the same state must have decreased by the same amount as the increase in the sum of the probabilities of transitions, conserving probability.

## 2.3 Expressing Transition Matrices in Terms of the Rate Matrices

We will now refer back to the Markov property to attempt to find an expression for the transition matrix. If we start by setting  $s = 0$  in our original transition matrix expression for the Markov

property,

$$\begin{aligned} M(0, t) &= M(0, u) M(u, t) \\ M(t) &= M(u) M(t - u). \end{aligned} \tag{2.10}$$

Now re-parametrising,

$$M(u + t) = M(u) M(t). \tag{2.11}$$

We derive the transition matrix from the definition of a derivative,

$$\begin{aligned} M'(t) &= \lim_{s \rightarrow 0} \frac{M(t + s) - M(t)}{s} \\ &= \lim_{s \rightarrow 0} \frac{M(t) M(s) - M(t)}{s} \\ &= M(t) \lim_{s \rightarrow 0} \frac{M(s) - M(0)}{s} \\ &= M(t) M'(0). \end{aligned} \tag{2.12}$$

We define the initial rate matrix to be the initial time derivative of the transition matrix,

$$Q := M'(0). \tag{2.13}$$

A solution to the differential equation is then

$$M(t) = Ae^{Qt}, \tag{2.14}$$

where  $A$  is some unknown  $n \times n$  matrix.

Initially,

$$M(0) = Ae^{Q \cdot 0} = A. \tag{2.15}$$

Since  $M(0) = I$ ,

$$M(t) = e^{Qt}. \tag{2.16}$$

Since  $Q$  is the time derivative of the transition matrix initially, it will be the matrix of the initial rates of transitions. Rates are fixed in time due to the homogeneity of the Markov chain. We can derive the transition matrix and utilise the conservation of probability to show that a



similar property holds for the rate matrix.

$$\begin{aligned}
\sum_{i=1}^n [M(t)]_{ij} &= 1 \text{ for all } j \\
\frac{d}{dt} \sum_{i=1}^n [M(t)]_{ij} &= 0 \\
\sum_{i=1}^n [M'(t)]_{ij} &= 0.
\end{aligned} \tag{2.17}$$

Since this property holds for all  $t$ , it must hold initially,

$$\begin{aligned}
\sum_{i=1}^n [M'(0)]_{ij} &= 0 \text{ for all } j \\
\sum_{i=1}^n Q_{ij} &= 0.
\end{aligned} \tag{2.18}$$

From the conservation of probability in the transition matrix, we can conclude that the sum of transition rates in the rate matrix must always be zero. Recalling that the off-diagonal elements of the transition matrix must increase over a short period of time after the initial time, we can show with a bit of manipulation that the off-diagonal elements of the rate matrix must be non-negative.

$$\begin{aligned}
[M(s)]_{ij} &\geq [M(0)]_{ij} \text{ for } i \neq j \\
\lim_{s \rightarrow 0} \frac{[M(s)]_{ij} - [M(0)]_{ij}}{s} &\geq 0 \\
Q_{ij} &\geq 0.
\end{aligned} \tag{2.19}$$

From a similar argument we find that the diagonal elements of the rate matrix must be non-positive, ensuring that the sum of the transition rates is zero. In summary,

$$\left\{ \begin{array}{l} \sum_{i=1}^n Q_{ij} = 0, \\ Q_{ij} \geq 0 \text{ for } i \neq j, \\ Q_{ii} \leq 0. \end{array} \right. \tag{2.20}$$

## 2.4 Transition Matrices as Matrix Exponentials

Since the number of current and final states is equal, the rate matrix must be square. This ensures that the transition matrix can be evaluated by taking a Taylor series expansion,

$$M(t) = e^{Qt} = I + Qt + \frac{Q^2 t^2}{2!} + \dots \tag{2.21}$$

In general, matrix exponentials are not easy to compute. If the matrix is diagonal, however, then matrix exponentiation is easy. The rate matrix,  $Q$ , is said to be *diagonalisable* if there exists some matrix,  $h$ , such that

$$\widehat{Q} = h^{-1}Qh = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n) \quad (2.22)$$

is a diagonal matrix with diagonal elements of  $\lambda_1$ ,  $\lambda_2$ , to  $\lambda_n$ .

The matrix exponentiation then simply becomes

$$\begin{aligned} e^{Qt} &= e^{h\widehat{Q}th^{-1}} \\ &= I + h\left(\widehat{Q}t\right)h^{-1} + \frac{h\left(\widehat{Q}t\right)^2h^{-1}}{2!} + \dots \\ &= hIh^{-1} + h\left(\widehat{Q}t\right)h^{-1} + h\frac{\left(\widehat{Q}t\right)^2}{2!}h^{-1} + \dots \\ &= he^{\widehat{Q}t}h^{-1} \\ &= he^{\text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)t}h^{-1} \\ &= h \cdot \text{diag}(e^{\lambda_1 t}, e^{\lambda_2 t}, \dots, e^{\lambda_n t}) \cdot h^{-1}, \end{aligned} \quad (2.23)$$

where  $\text{diag}(e^{\lambda_1 t}, e^{\lambda_2 t}, \dots, e^{\lambda_n t})$  is a diagonal matrix with the diagonal elements being  $e^{\lambda_1 t}$ ,  $e^{\lambda_2 t}$ , to  $e^{\lambda_n t}$ .

We will now discuss some models that are diagonalisable.

## 2.5 The Abelian Group-Based Models

Suppose the permutation matrices,  $\{I, K_1, K_2, \dots, K_m\}$  under matrix multiplication are isomorphic to an Abelian group. We will then refer to the model as being an *Abelian group-based* model or simply a *group-based* model, as defined by Semple and Steel [2003]. If a model is Abelian group-based then it has a one dimensional irreducible representation. This means that the transformed rate matrix,  $\widehat{Q}$ , can be expressed as a block matrix, with the off-diagonal blocks being the zero matrices (one dimensional) and the diagonal blocks being the scalar (one dimensional) eigenvalues of the rate matrix. The transformed rate matrix can then be expressed as

$$\widehat{Q} = h^{-1}Qh = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n). \quad (2.24)$$

In other words, if a model is Abelian group-based then it is diagonalisable and the matrix exponential is easy to find. We will discuss how to diagonalise Abelian group-based models and provide some examples. Some of these examples can be found in Sumner et al. [2014].

### 2.5.1 Diagonalising the Abelian Group-Based Models

Suppose we have an Abelian group-based model that we wish to diagonalise. To diagonalise our model we need to find its eigenvectors. We start by expressing the rate matrix as a linear combination of Markov generators,

$$Q = \sum_{i=1}^k \alpha_i L_i, \quad (2.25)$$

where  $L_i$  are the  $k$  Markov generators, each with a rate parameter,  $\alpha_i$ .

The Markov generators are the set of Markov matrices that span the space of the Markov model. For example, for the binary symmetric model there is only one rate parameter and consequently only one Markov generator is required.

Furthermore, we can express the Markov generators as linear combinations of the elements of the Abelian group. This then allows us to express the rate matrix as a linear combination of the group elements,

$$Q = \sum_{i=1}^{n-1} \alpha_i (-I + K_i), \quad (2.26)$$

where  $n - 1 \geq k$  since rate parameters for different group elements may be equal.

We now use the Cayley-Hamilton theorem to find the eigenvectors and eigenvalues of an element of the Abelian group by finding the minimal polynomial of the group element from its characteristic polynomial. Each factor of the characteristic polynomial is an eigenvalue of the group element subtracted from the group element itself. The degree of each factor of the characteristic polynomial is the number of eigenvectors that share that eigenvalue.

From the Cayley-Hamilton theorem, the characteristic polynomial of each group element, including  $I$ , must be identically zero,

$$\begin{aligned} p(K_i) &= K_i^n + c_{n-1}K_i^{n-1} + \dots + c_1K_i + c_0I = 0 \\ (K_i - \lambda_1 I)^{r_1} (K_i - \lambda_2 I)^{r_2} \dots (K_i - \lambda_m I)^{r_m} &= 0, \end{aligned} \quad (2.27)$$

So  $\lambda_1, \lambda_2, \dots, \lambda_m$  are the  $m$  distinct eigenvalues of  $K_i$ . We can conclude that the eigenvectors and eigenvalues of  $K_i$  can be found by finding the minimal polynomial of  $K_i$ , the lowest degree polynomial such that  $p(K_i) = 0$ . The factors in the minimal polynomial are the same as those in the characteristic polynomial. The degree of each factor in the characteristic polynomial is the number of eigenvectors with that eigenvalue, whereas the degree of every factor in the minimal polynomial is one. The minimal polynomial, therefore, does not tell us how many eigenvectors share a given eigenvalue.

Now suppose a group element,  $K_i$ , has a unique eigenvector,  $v$ , and an eigenvalue,  $\lambda$ . The

equation relating the group element to the eigenvector and the eigenvalue is

$$K_i v = \lambda v. \quad (2.28)$$

For an Abelian group, any two elements commute,

$$[K_i, K_j] := K_i K_j - K_j K_i = 0. \quad (2.29)$$

As a consequence, for two arbitrary group elements,  $K_i$  and  $K_j$ ,

$$K_i K_j v = K_j K_i v = K_j \lambda v = \lambda K_j v. \quad (2.30)$$

We conclude that  $K_j v$  is also an eigenvector of  $K_i$ , with the same eigenvalue,  $\lambda$ .

The degree of the characteristic polynomial is equal to the number of states in the state space of the Markov model,  $n$ . If the degree of the minimal polynomial for  $K_i$  is equal to the degree of the characteristic polynomial for  $K_i$ , then the  $n$  eigenvectors of  $K_i$  must all have distinct eigenvalues. These  $n$  eigenvectors must be eigenvectors of all of the elements of the group.

Suppose now that the degree of the minimal polynomial for  $K_i$  is less than the degree of the characteristic polynomial. Some of the eigenvectors must belong to a multi-dimensional eigenspace. In other words, there are multiple eigenvectors with the same eigenvalue. We can use this fact to find the remaining eigenvectors of  $K_j$ . We continue this process if necessary to find the eigenvectors of all of the elements of the group. Alternatively, we can first find the minimal polynomials for all of the group elements. If at least one of the minimal polynomials of the group elements has degree  $n$ , the eigenvectors of these group elements must all be eigenvectors of every element of the group.

Supposing we have now found the set of eigenvectors for all of the group elements, we can multiply linear combinations of the group elements by each eigenvector and show that these eigenvectors are also eigenvectors of the entire model,

$$\begin{aligned} \left[ \sum_{i=1}^{n-1} \alpha_i (-I + K_i) \right] v &= \left[ \sum_{i=1}^{n-1} \alpha_i (-1 + \lambda_i) \right] v \\ Qv &= \left[ \sum_{i=1}^{n-1} \alpha_i (-1 + \lambda_i) \right] v, \end{aligned} \quad (2.31)$$

where the eigenvalue of  $I$  is one and the  $\lambda_i$  are the eigenvalues of the  $K_i$ .

Now that we have the set of eigenvectors for the Markov model, the matrix formed by the set of these eigenvectors will diagonalise the model.

An appropriate exact method in our context of finding the minimal polynomial of the group element is to start with the lowest possible degree polynomial and increase the degree by one at a time until the polynomial becomes identically zero. This is a sufficient method for our needs, but it is only feasible without resorting to numerical methods because the degree of the minimal

polynomials are at most the dimensions of the Markov models in question. For example, for an  $n$  state Markov model, the degree of the minimal polynomial must be no greater than  $n$ . Apart from the identity element, since the group element will never be the zero matrix or a matrix proportional to the identity matrix, the degree of the minimal polynomial must be at least two. To find the minimal polynomial we start by squaring the group element. If the square of the group element can be expressed in terms of just the group element itself and the identity matrix, then we have found the minimal polynomial. If this cannot be done (i.e. the square of the group element must be expressed in terms of the other non-identity group elements) then the degree of the minimal polynomial must be at least three. We then proceed to find the cube of the group element and repeat the process until we find the minimal polynomial.

To illustrate how to find the eigenvectors and eigenvalues from the minimal polynomial, let's look at a minimal polynomial as an example. An arbitrary minimal polynomial can be expressed as

$$(K_i - \lambda_1 I)(K_i - \lambda_2 I) \dots (K_i - \lambda_m I) = 0. \quad (2.32)$$

Since this is the minimal polynomial, there exists a vector,  $v \in \mathbb{C}^n$ , such that

$$u := [(K_i - \lambda_2 I)(K_i - \lambda_3 I) \dots (K_i - \lambda_m I)] v \neq 0. \quad (2.33)$$

Therefore,

$$\begin{aligned} K_i u &= K_i [(K_i - \lambda_2 I)(K_i - \lambda_3 I) \dots (K_i - \lambda_m I)] v \\ &= \lambda_i [(K_i - \lambda_2 I)(K_i - \lambda_3 I) \dots (K_i - \lambda_m I)] v, \end{aligned} \quad (2.34)$$

where

$$[(K_i - \lambda_2 I)(K_i - \lambda_3 I) \dots (K_i - \lambda_m I)] v \quad (2.35)$$

is an eigenvector of  $K_i$ .

Since the factors of the minimal polynomial commute we can use the appropriate orderings on the factors of the minimal polynomial to find all of the eigenvectors of  $K_i$ .

### 2.5.2 The Binary Symmetric Model

We will now look at diagonalising the binary symmetric model,  $\mathbb{Z}_2 \cong \{e, (12)\}$ , as an example. The binary symmetric model is also known in the literature as the Cavender-Farris-Neyman (CFN) model or the Neyman 2-state model. More generally, the  $r$  state model with the same substitution rate for every substitution is known as the  $N_r$  model. The  $N_4$  model is more commonly known as the Jukes-Cantor (JC) model.

The rate matrix for the binary symmetric model is

$$Q = \begin{bmatrix} -\alpha & \alpha \\ \alpha & -\alpha \end{bmatrix}, \quad (2.36)$$

where  $\alpha > 0$  is the rate of mutations between the two states in the state space.

The rate matrix can be written as

$$Q = \alpha L = \alpha (-I + K), \quad (2.37)$$

where

$$K = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \text{ with } K^2 = I. \quad (2.38)$$

Clearly the group formed by the set  $\{I, K\}$  under matrix multiplication is isomorphic to  $\mathbb{Z}_2 \cong \{e, (12)\}$ , the cyclic group of order 2, where  $e$  is the identity permutation and  $(12)$  is the permutation of the objects “1” and “2”. This means that the binary symmetric model is an Abelian group-based model.

We will now proceed to diagonalise the binary symmetric model. We have already argued that  $K^2 = I$ , therefore the minimal polynomial for  $K$  must be

$$K^2 - I = 0. \quad (2.39)$$

Factorising the minimal polynomial,

$$K^2 - I = (K - I)(K + I) = 0. \quad (2.40)$$

The degree of the minimal polynomial is two and there are two distinct eigenvalues. We can conclude that the minimal polynomial must also be the characteristic polynomial and the eigenvectors of  $K$  will diagonalise the model. Since  $(K - I)(K + I)$  is the minimal polynomial, there exists a vector,  $v \in \mathbb{C}^2$ , such that

$$u := (K + I)v \neq 0. \quad (2.41)$$

Therefore

$$Ku = K(K + I)v = (K + I)v = \left\{ k_1 \begin{bmatrix} 1 \\ 1 \end{bmatrix}, k_1 \in \mathbb{C} \setminus \{0\} \right\}. \quad (2.42)$$

Similarly, there exists a vector,  $v' \in \mathbb{C}^2$ , such that

$$u' := (K - I)v' \neq 0. \quad (2.43)$$

Therefore

$$Ku' = K(K - I)v' = -(K - I)v' = \left\{ k_2 \begin{bmatrix} 1 \\ -1 \end{bmatrix}, k_2 \in \mathbb{C} \setminus \{0\} \right\}. \quad (2.44)$$

These eigenvectors, in union with the zero vector, then form an eigenspace for  $K$ , thus diagonalising it. The diagonalising matrix will then be

$$h = \begin{bmatrix} k_1 & k_2 \\ k_1 & -k_2 \end{bmatrix}, \quad (2.45)$$

where the determinant must be non-zero by definition. For this diagonalising matrix, the determinant will be  $2k_1k_2$ , which implies that  $k_1, k_2 \in \mathbb{C} \setminus \{0\}$ .

The diagonalised rate matrix for the binary symmetric model will then be

$$\hat{Q} = \begin{bmatrix} 0 & 0 \\ 0 & -2\alpha \end{bmatrix}. \quad (2.46)$$

### 2.5.3 The $\mathbb{Z}_3$ Model

We will now look at the  $\mathbb{Z}_3$  model,  $\mathbb{Z}_3 \cong \{e, (123), (132)\}$ , as a slightly more complicated example. The rate matrix for the model is

$$\begin{aligned} Q &= \begin{bmatrix} -(\alpha + \beta) & \alpha & \beta \\ \beta & -(\alpha + \beta) & \alpha \\ \alpha & \beta & -(\alpha + \beta) \end{bmatrix} \\ &= \alpha L_\alpha + \beta L_\beta, \end{aligned} \quad (2.47)$$

where

$$L_\alpha = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ 1 & 0 & -1 \end{bmatrix}, \quad L_\beta = \begin{bmatrix} -1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix} \quad (2.48)$$

are the zero column sum Markov generators.

In terms of the elements of the Abelian group,  $\mathbb{Z}_3$ , we can express the Markov generators as

$$L_\alpha = -I + K_\alpha, \quad L_\beta = -I + K_\beta, \quad (2.49)$$

where

$$K_\alpha = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad K_\beta = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \quad (2.50)$$

It can easily be shown that

$$\begin{cases} K_\alpha^2 &= K_\beta, \\ K_\alpha^3 &= I, \end{cases} \quad (2.51)$$

confirming the isomorphism to the  $\mathbb{Z}_3$  group.

Since  $K_\alpha^2 = K_\beta$ , the square of  $K_\alpha$  must be written in terms of the other elements of the group. The degree of the minimal polynomial must therefore be at least three. Since  $K_\alpha^3 = I$ , the minimal polynomial must be

$$K_\alpha^3 - I = 0. \quad (2.52)$$

Factorising the minimal polynomial,

$$K_\alpha^3 - I = (K_\alpha - I)(K_\alpha - \omega I)(K_\alpha - \omega^2 I) = 0, \quad (2.53)$$

where

$$\omega = e^{i\frac{2\pi}{3}}. \quad (2.54)$$

Since the degree of the minimal polynomial for  $K_\alpha$  is three, and each eigenvector has a distinct eigenvalue, the minimal polynomial must also be the characteristic polynomial. The eigenvector for  $K_\alpha$  must therefore diagonalise the model and it is not necessary to find the minimal polynomial for the other non-trivial (non-identity) element of the group.

Following the same argument used for the binary symmetric model we find that there are some vectors,  $v_1$ ,  $v_2$  and  $v_3$ , such that the eigenvectors for  $K_\alpha$  satisfy

$$\begin{aligned} K_\alpha [(K_\alpha - \omega I)(K_\alpha - \omega^2 I)] v_1 &= [(K_\alpha - \omega I)(K_\alpha - \omega^2 I)] v_1 = \left\{ k_1 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, k_1 \in \mathbb{C} \setminus \{0\} \right\}, \\ K_\alpha [(K_\alpha - I)(K_\alpha - \omega^2 I)] v_2 &= \omega [(K_\alpha - I)(K_\alpha - \omega^2 I)] v_2 = \left\{ k_2 \begin{bmatrix} 1 \\ \omega \\ \omega^2 \end{bmatrix}, k_2 \in \mathbb{C} \setminus \{0\} \right\}, \\ K_\alpha [(K_\alpha - I)(K_\alpha - \omega I)] v_3 &= \omega^2 [(K_\alpha - I)(K_\alpha - \omega I)] v_3 = \left\{ k_3 \begin{bmatrix} 1 \\ \omega^2 \\ \omega \end{bmatrix}, k_3 \in \mathbb{C} \setminus \{0\} \right\}. \end{aligned} \quad (2.55)$$

These eigenvectors, in union with the zero vector, then form an eigenspace for  $K_\alpha$  and in



turn for the model itself, thus diagonalising it. We then find the diagonalising matrix to be

$$h = \begin{bmatrix} k_1 & k_2 & k_3 \\ k_1 & k_2 e^{i\frac{2\pi}{3}} & k_3 e^{-i\frac{2\pi}{3}} \\ k_1 & k_2 e^{-i\frac{2\pi}{3}} & k_3 e^{i\frac{2\pi}{3}} \end{bmatrix}, \quad (2.56)$$

where the determinant must be non-zero by definition, which implies that  $k_1, k_2, k_3 \in \mathbb{C} \setminus \{0\}$ .

The diagonalised rate matrix will then be

$$\hat{Q} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{3(\beta-\alpha\omega)}{-1+\omega} & 0 \\ 0 & 0 & \frac{3(\alpha-\beta\omega)}{-1+\omega} \end{bmatrix}. \quad (2.57)$$

#### 2.5.4 The K3ST Model

For a further example, we will now look at the K3ST model,

$$\mathbb{Z}_2 \times \mathbb{Z}_2 \cong \{e, (12)(34), (13)(24), (14)(23)\}. \quad (2.58)$$

The rate matrix for the model is

$$Q = \begin{bmatrix} -(\alpha + \beta + \gamma) & \alpha & \beta & \gamma \\ \alpha & -(\alpha + \beta + \gamma) & \gamma & \beta \\ \beta & \gamma & -(\alpha + \beta + \gamma) & \alpha \\ \gamma & \beta & \alpha & -(\alpha + \beta + \gamma) \end{bmatrix} \quad (2.59)$$

$$= \alpha L_\alpha + \beta L_\beta + \gamma L_\gamma,$$

where

$$L_\alpha = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}, \quad L_\beta = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}, \quad (2.60)$$

$$L_\gamma = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix}$$

are the zero column sum Markov generators.

In terms of the elements of the Abelian group,  $\mathbb{Z}_2 \times \mathbb{Z}_2$ , we can express the Markov generators

as

$$L_\alpha = -I + K_\alpha, \quad L_\beta = -I + K_\beta, \quad L_\gamma = -I + K_\gamma \quad (2.61)$$

where

$$K_\alpha = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad K_\beta = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad K_\gamma = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \quad (2.62)$$

For all  $i, j, k \in \{\alpha, \beta, \gamma\}$ , with all subscripts being different elements, it can easily be shown that

$$\begin{cases} K_i K_j = K_j K_i = K_k, \\ K_i^2 = I, \end{cases} \quad (2.63)$$

confirming the isomorphism to the  $\mathbb{Z}_2 \times \mathbb{Z}_2$  group.

We can immediately see the minimal polynomials from the group structure. Note the similarity of the minimal polynomials for the K3ST model, which is isomorphic to  $\mathbb{Z}_2 \times \mathbb{Z}_2$ , to the minimal polynomial for the binary symmetric model, which is isomorphic to the  $\mathbb{Z}_2$  group. These minimal polynomials will therefore have the same eigenvalues as each other and can be factorised as

$$K_i^2 - I = (K_i - I)(K_i + I) = 0. \quad (2.64)$$

Following the same argument used for the previous examples we find that there are some vectors,  $v_1$  and  $v_2$ , such that the eigenvectors for  $K_\alpha$  satisfy

$$\begin{cases} K_\alpha [(K_\alpha + I)] v_1 = [(K_\alpha + I)] v_1, \\ K_\alpha [(K_\alpha - I)] v_2 = -[(K_\alpha - I)] v_2. \end{cases} \quad (2.65)$$

Recall that if some vector,  $v$ , is an eigenvector of some group element,  $K_i$ , with eigenvalue,  $\lambda$ , then  $v$  is also an eigenvector of some other group element,  $K_j$ , with eigenvalue,  $\mu$ . If

$$K_\alpha v = \lambda v, \quad (2.66)$$

then

$$K_\alpha K_\beta v = K_\beta K_\alpha v = K_\beta \lambda v = \lambda K_\beta v. \quad (2.67)$$

Adding linear combinations of these equations

$$K_\alpha (K_\beta - \mu I) v = \lambda (K_\beta - \mu I) v. \quad (2.68)$$

For  $(K_\beta - \mu I) v$  to also be the eigenvectors of  $K_\beta$ , we require

$$K_\beta (K_\beta - \mu I) v = \nu (K_\beta - \mu I) v. \quad (2.69)$$

From the same argument as that used for  $K_\alpha$ , these equations are satisfied when  $\mu = \pm 1$ , giving

$$\begin{cases} K_\beta (K_\beta + I) v &= (K_\beta + I) v, \\ K_\beta (K_\beta - I) v &= -(K_\beta - I) v. \end{cases} \quad (2.70)$$

Applying this result to our original equations for the eigenvectors of  $K_\alpha$ ,

$$\begin{cases} K_\alpha [(K_\beta - \mu I) (K_\alpha + I)] v_1 &= [(K_\beta - \mu I) (K_\alpha + I)] v_1, \\ K_\alpha [(K_\beta - \mu I) (K_\alpha - I)] v_2 &= -[(K_\beta - \mu I) (K_\alpha - I)] v_2. \end{cases} \quad (2.71)$$

The four eigenvectors are then

$$\begin{aligned} K_\alpha [(K_\beta + I) (K_\alpha + I)] v_1 &= [(K_\beta + I) (K_\alpha + I)] v_1 = \left\{ k_1 \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, k_1 \in \mathbb{C} \setminus \{0\} \right\}, \\ K_\alpha [(K_\beta + I) (K_\alpha - I)] v_2 &= [(K_\beta + I) (K_\alpha - I)] v_2 = \left\{ k_2 \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}, k_2 \in \mathbb{C} \setminus \{0\} \right\}, \\ K_\alpha [(K_\beta - I) (K_\alpha + I)] v_1 &= -[(K_\beta - I) (K_\alpha + I)] v_1 = \left\{ k_3 \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}, k_3 \in \mathbb{C} \setminus \{0\} \right\}, \\ K_\alpha [(K_\beta - I) (K_\alpha - I)] v_2 &= -[(K_\beta - I) (K_\alpha - I)] v_2 = \left\{ k_4 \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix}, k_4 \in \mathbb{C} \setminus \{0\} \right\}. \end{aligned} \quad (2.72)$$

These eigenvectors, in union with the zero vector, then form an eigenspace for  $K_\alpha$  and  $K_\beta$ . Since the eigenspace is of rank four it must diagonalise the model. We then find the diagonalising

matrix to be

$$h = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \\ k_1 & -k_2 & k_3 & -k_4 \\ k_1 & k_2 & -k_3 & -k_4 \\ k_1 & -k_2 & -k_3 & k_4 \end{bmatrix}, \quad (2.73)$$

where the determinant must be non-zero by definition, which implies that  $k_1, k_2, k_3, k_4 \in \mathbb{C} \setminus \{0\}$ .

The diagonalised rate matrix will then be

$$\hat{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -2(\alpha + \gamma) & 0 & 0 \\ 0 & 0 & -2(\beta + \gamma) & 0 \\ 0 & 0 & 0 & -2(\alpha + \beta) \end{bmatrix}. \quad (2.74)$$

## 2.6 Finding Diagonalising Matrices in *Mathematica*

*Mathematica* has a function called “Eigenvectors”. This function will output a set of eigenvectors for our rate matrix,  $Q$ , of a given Abelian group-based model or, more generally, for any diagonalisable matrix. This set of eigenvectors can be used to diagonalise the rate matrix. Unfortunately the function does not output the most general set of eigenvectors for a given matrix. The code below can be used in *Mathematica* to find the diagonalising matrix formed by the most general set of eigenvectors.

```
(* The rate matrix, Q, of an Abelian group-based model. Qij is the
   element in the i^th row and j^th column of the rate matrix. *)
Q = {{Q11, Q12, ..., Q1n}, {Q21, Q22, ..., Q2n}, ..., {Qn1, Qn2, ...,
   Qnn}};

(* n is the number of columns and rows in Q and can be taken to be
   the dimension of any row or column in Q. Here n is the dimension
   of the first row of Q. *)
n = Dimensions[Q][[1]][[1]];

(* An n*n table with arbitrary elements. Each column of the table
   will be an eigenvector of Q. *)
K = Table[k[i, j], {i, n}, {j, n}];

(* The eigenvalues of Q. *)
L = lambda /. FullSimplify[Solve[{Det[Q - lambda*IdentityMatrix[n]]
   == 0}, {lambda}]];
```

```
(* The diagonalising matrix formed by the eigenvectors of Q. *)
h = MatrixForm[Transpose[Table[K[[All, i]] /. FullSimplify[Solve[{Q.K
  [[All, i]] == L[[i]]*K[[All, i]]}, K[[All, i]]]], {i, n}]]]
```

For example, for the binary symmetric model, we would take  $Q$  to be

$$Q = \{\{-\alpha, \alpha\}, \{\alpha, -\alpha\}\};$$

The diagonalising matrix output for the binary symmetric model is shown below.

$$\left( \begin{pmatrix} k[1,1] \\ k[1,1] \end{pmatrix} \begin{pmatrix} k[1,2] \\ -k[1,2] \end{pmatrix} \right)$$

## 2.7 The Four-State General Markov Model

We have shown that Abelian group-based models are diagonalisable and provided some examples. This raises the question as to whether models that are not Abelian group-based can be diagonalised. Although having an isomorphism to an Abelian group is a sufficient condition for a Markov model to be diagonalisable, it is not a necessary condition. There are some models which are not Abelian group-based, which are nonetheless diagonalisable. The lowest order group which is not Abelian is the  $\mathfrak{S}_3$  group, which has order six. Despite not being an Abelian group-based model, the  $\mathfrak{S}_3$  group-based model can be diagonalised. For an example of a model which is not Abelian group-based, we will look at the general Markov model for four states. The rate matrix is

$$Q = \sum_{i=1, j \neq i}^n \alpha_{ij} L_{ij}, \quad (2.75)$$

where  $n$  is the number of states in the state space,  $\alpha_{ij} > 0$  are the rate parameters and  $L_{ij}$  are the generators. In Sumner et al. [2012b] it was shown that the commutators for the irreducible representations of the generators satisfy

$$[L_{ij}, L_{kl}] = (L_{il} - L_{jl})(\delta_{jk} - \delta_{jl}) - (L_{kj} - L_{lj})(\delta_{il} - \delta_{jl}), \quad (2.76)$$

where  $\delta_{ij}$  is the Kronecker delta.

Since the commutator is not generally equal to zero, we can conclude that the general Markov model is not Abelian group-based. Therefore it cannot be assumed that the general Markov model can be diagonalised. Indeed, it is not possible to diagonalise the general Markov model.

## 2.8 Abelian Group-Based Models and Their Diagonalising Matrices

We can follow a similar process to what was used for the binary symmetric model, the  $\mathbb{Z}_3$  model and the K3ST model to diagonalise any Abelian group-based model. Refer to the *Mathematica* code in Section 2.6 to see how we found the diagonalising matrices for the remaining Abelian group-based models. Here we have swapped the order of some of the eigenvectors in the diagonalising matrices and relabelled the parameters. Below in Table 2.1 is a list of the rate matrices and diagonalising matrices for Abelian group-based models.

Model	Rate Matrix ( $Q$ )	Diagonalising Matrix ( $h$ )
Binary Symmetric ( $\mathbb{Z}_2$ )	$\begin{bmatrix} * & \alpha \\ \alpha & * \end{bmatrix}$	$\begin{bmatrix} k_1 & k_2 \\ k_1 & -k_2 \end{bmatrix}$
$\mathbb{Z}_3$	$\begin{bmatrix} * & \alpha & \beta \\ \beta & * & \alpha \\ \alpha & \beta & * \end{bmatrix}$	$\begin{bmatrix} k_1 & k_2 & k_3 \\ k_1 & k_2 e^{i\frac{2\pi}{3}} & k_3 e^{-i\frac{2\pi}{3}} \\ k_1 & k_2 e^{-i\frac{2\pi}{3}} & k_3 e^{i\frac{2\pi}{3}} \end{bmatrix}$
Three State Neyman ( $\alpha = \beta$ ) ( $\mathbb{Z}_3$ )	$\begin{bmatrix} * & \alpha & \alpha \\ \alpha & * & \alpha \\ \alpha & \alpha & * \end{bmatrix}$	$\begin{bmatrix} k_1 & k_2 & k_4 \\ k_1 & k_3 & k_5 \\ k_1 & -(k_2 + k_3) & -(k_4 + k_5) \end{bmatrix}$
$\mathbb{Z}_2 \times \mathbb{Z}_2 \cong$ $\{e, (12), (34), (12)(34)\}$	$\begin{bmatrix} * & \alpha & 0 & 0 \\ \alpha & * & 0 & 0 \\ 0 & 0 & * & \beta \\ 0 & 0 & \beta & * \end{bmatrix}$	$\begin{bmatrix} k_1 & k_2 & k_4 & 0 \\ -k_1 & k_2 & k_4 & 0 \\ 0 & k_3 & k_5 & k_6 \\ 0 & k_3 & k_5 & -k_6 \end{bmatrix}$
$(\alpha = \beta)$ ( $\mathbb{Z}_2 \times \mathbb{Z}_2 \cong$ $\{e, (12), (34), (12)(34)\}$ )	$\begin{bmatrix} * & \alpha & 0 & 0 \\ \alpha & * & 0 & 0 \\ 0 & 0 & * & \alpha \\ 0 & 0 & \alpha & * \end{bmatrix}$	$\begin{bmatrix} k_1 & k_3 & k_5 & k_7 \\ k_1 & -k_3 & k_5 & -k_7 \\ k_2 & k_4 & k_6 & k_8 \\ k_2 & -k_4 & k_6 & -k_8 \end{bmatrix}$
K3ST ( $\mathbb{Z}_2 \times \mathbb{Z}_2 \cong$ $\{e, (12)(34), (13)(24), (14)(23)\}$ )	$\begin{bmatrix} * & \alpha & \beta & \gamma \\ \alpha & * & \gamma & \beta \\ \beta & \gamma & * & \alpha \\ \gamma & \beta & \alpha & * \end{bmatrix}$	$\begin{bmatrix} k_1 & k_2 & k_3 & k_4 \\ k_1 & -k_2 & k_3 & -k_4 \\ k_1 & k_2 & -k_3 & -k_4 \\ k_1 & -k_2 & -k_3 & k_4 \end{bmatrix}$
K2ST ( $\alpha = \beta$ ) ( $\mathbb{Z}_2 \times \mathbb{Z}_2 \cong$ $\{e, (12)(34), (13)(24), (14)(23)\}$ )	$\begin{bmatrix} * & \alpha & \alpha & \gamma \\ \alpha & * & \gamma & \alpha \\ \alpha & \gamma & * & \alpha \\ \gamma & \alpha & \alpha & * \end{bmatrix}$	$\begin{bmatrix} k_1 & k_2 & k_4 & k_6 \\ k_1 & -k_3 & k_5 & -k_6 \\ k_1 & k_3 & -k_5 & -k_6 \\ k_1 & -k_2 & -k_4 & k_6 \end{bmatrix}$
$\mathbb{Z}_4 \cong \{e, (12)(34), (1324), (1423)\}$	$\begin{bmatrix} * & \alpha & \beta & \gamma \\ \alpha & * & \gamma & \beta \\ \gamma & \beta & * & \alpha \\ \beta & \gamma & \alpha & * \end{bmatrix}$	$\begin{bmatrix} k_1 & k_2 & k_3 & k_4 \\ k_1 & -k_2 & k_3 & -k_4 \\ k_1 & ik_2 & -k_3 & -ik_4 \\ k_1 & -ik_2 & -k_3 & ik_4 \end{bmatrix}$

2 Parameter $\mathbb{Z}_4$ ( $\alpha = \beta$ ) ( $\mathbb{Z}_4 \cong \{e, (12)(34), (1324), (1423)\}$ )	$\begin{bmatrix} * & \alpha & \alpha & \gamma \\ \alpha & * & \gamma & \alpha \\ \gamma & \alpha & * & \alpha \\ \alpha & \gamma & \alpha & * \end{bmatrix}$	$\begin{bmatrix} k_1 & k_2 & k_3 & k_4 \\ k_1 & -k_2 & k_3 & -k_4 \\ k_1 & ik_2 & -k_3 & -ik_4 \\ k_1 & -ik_2 & -k_3 & ik_4 \end{bmatrix}$
2 Parameter $\mathbb{Z}_4$ ( $\beta = \gamma$ ) ( $\mathbb{Z}_4 \cong \{e, (12)(34), (1324), (1423)\}$ )	$\begin{bmatrix} * & \alpha & \beta & \beta \\ \alpha & * & \beta & \beta \\ \beta & \beta & * & \alpha \\ \beta & \beta & \alpha & * \end{bmatrix}$	$\begin{bmatrix} k_1 & k_2 & k_4 & k_5 \\ k_1 & -k_2 & k_4 & -k_5 \\ k_1 & k_3 & -k_4 & -k_6 \\ k_1 & -k_3 & -k_4 & k_6 \end{bmatrix}$
JC ( $\mathbb{Z}_2 \times \mathbb{Z}_2$ or $\mathbb{Z}_4$ )	$\begin{bmatrix} * & \alpha & \alpha & \alpha \\ \alpha & * & \alpha & \alpha \\ \alpha & \alpha & * & \alpha \\ \alpha & \alpha & \alpha & * \end{bmatrix}$	$\begin{bmatrix} k_1 & k_2 & k_5 & k_8 \\ k_1 & k_3 & k_6 & k_9 \\ k_1 & k_4 & k_7 & k_{10} \\ k_1 & * & * & * \end{bmatrix}$

Table 2.1: The Abelian group-based models for two to four states and their diagonalising matrices. Each  $*$  is the negative of the sum of all other elements in that column. Every parameter in the diagonalising matrices can be an arbitrary, non-zero complex number. The parameters of the diagonalising matrices must all be constrained to have non-zero determinants and are therefore invertible.

We have left out some models that are equivalent to the models shown up to permutation of the elements of the state space. For example, suppose we took the rate matrix for the  $\mathbb{Z}_4 \cong \{e, (12)(34), (1324), (1423)\}$  model and labelled the rows and columns as shown below,

$$Q = \left[ \begin{array}{c|cccc} & A & G & C & T \\ \hline A & * & \alpha & \beta & \gamma \\ G & \alpha & * & \gamma & \beta \\ C & \gamma & \beta & * & \alpha \\ T & \beta & \gamma & \alpha & * \end{array} \right], \quad (2.77)$$

where  $Q_{ij}$  is the rate of substitutions from state  $j$  to state  $i$ .

If we swapped the row labels for  $G$  and  $C$  then the rate matrix would become

$$Q = \left[ \begin{array}{c|cccc} & A & C & G & T \\ \hline A & * & \beta & \alpha & \gamma \\ C & \gamma & * & \beta & \alpha \\ G & \alpha & \gamma & * & \beta \\ T & \beta & \alpha & \gamma & * \end{array} \right]. \quad (2.78)$$

Clearly the rate matrix is displaying the same substitution rates, however it is now for the  $\mathbb{Z}_4 \cong \{e, (13)(24), (1234), (1432)\}$  isomorphism. Hence we do not need to consider the  $\mathbb{Z}_4 \cong \{e, (13)(24), (1234), (1432)\}$  isomorphism. Similarly, we have not considered the 2 Parameter

$\mathbb{Z}_4$  ( $\beta = \gamma$ ) ( $\mathbb{Z}_4 \cong \{e, (12)(34), (1324), (1423)\}$ ) model.

It should be noted that some of these models are submodels of other models. For example, the four state JC model is a submodel of the  $\mathbb{Z}_2 \times \mathbb{Z}_2 \cong \{e, (12)(34), (13)(24), (14)(23)\}$  isomorphism of the K3ST model. If we set  $\alpha = \beta = \gamma$  in this isomorphism of the K3ST model then we get the JC model. The diagonalising matrix for the isomorphism of the K3ST model diagonalises the model for *all* parameter choices. Consequently, since the JC model is a parameter choice of the isomorphism of the K3ST model, the JC model will be diagonalised by the diagonalising matrix of the isomorphism of the K3ST model. There will, however, be other choices for the diagonalising matrix for the JC model that do not diagonalise the isomorphism of the K3ST model.

Bryant [2005] suggested a method of generating rate matrices from Abelian group-based models. If the process is ergodic then the stationary distribution will be the uniform distribution. Furthermore, if the process is time reversible then the rate matrix will be symmetric. Imposing the symmetry condition on the rate matrix for  $\mathbb{Z}_4$  gives the K2ST model. Since the matrix representations of  $\mathbb{Z}_2 \times \mathbb{Z}_2$  must be symmetrical,  $\mathbb{Z}_2 \times \mathbb{Z}_2$  generates the K3ST model.

## 2.9 Choosing the Diagonalising Matrix Parameters

For every diagonalising matrix, we have some degrees of freedom. We will argue that some of these choices are more appropriate than others, particularly the choices that lead us to the statement of the conservation of probability. With the exception of the  $\mathbb{Z}_2 \times \mathbb{Z}_2 \cong \{e, (12), (34), (12)(34)\}$  isomorphism of the K3ST model, setting every element of the first row of every diagonalising matrix to one sets the first element of the transformed phylogenetic tensor to be the statement of the conservation of probability. It should also be noted that this results in the diagonalising matrices for the binary symmetric model and the K3ST ( $\mathbb{Z}_2 \times \mathbb{Z}_2 \cong \{e, (12)(34), (13)(24), (14)(23)\}$ ) model being the Hadamard matrices of the appropriate orders. If we choose the appropriate parameters for the diagonalising matrices for the K2ST models, some of the 2 Parameter  $\mathbb{Z}_4$  models and the JC model, then we again get the Hadamard matrix. Clearly the Hadamard matrix is important in phylogenetic analysis. As stated earlier, the Hadamard matrix provides the statement of conservation of probability for many Abelian group-based models since every element of the first row is one. By definition, every element of every row of a Hadamard matrix must be  $\pm 1$  and each row must be orthogonal to every other row. Consequently, exactly half the elements of every row other than the first row must be 1 and the other half must be  $-1$ . Other than the first element representing the conservation of probability, every other element of the transformed phylogenetic tensor must represent the difference in two sums of probabilities, with each sum containing the same number of elements.



## Chapter 3

# Phylogenetic Tensors, the Splitting Operator and Convergence

The following chapters of this thesis will focus on the analysis of various tree and network structures under Markov models of evolution. A framework for phylogenetic tensors will be crucial to our comparisons of the trees and networks. Phylogenetic tensors are functions of the rate parameters from the model in question and the time parameters from the tree or network. Phylogenetic tensors express the theoretical probabilities for each combination of states in the character space across the set of taxa. For a given number of taxa, we can compare the phylogenetic tensors for various trees and networks. For a given tree or network structure, we can compare all of the elements of the phylogenetic tensor to determine a set of equality and inequality constraints on a phylogenetic tensor. This set of constraints will correspond to a probability space that the phylogenetic tensor lies in. We can then compare the probability spaces for various trees and networks.

### 3.1 Character Sequence Alignments

Suppose we have an  $n$ -taxon character sequence. The state space has  $m$  states. These  $m$  states are labelled  $0, 1, \dots, m-1$ . At each site in the character sequence, each taxon takes one state from the state space. The combination of states,  $\mathbf{i}$ , at a site in the sequence alignment can be identified with  $\mathbf{i} \equiv i_1 i_2 \dots i_n$ , where each subscript refers to the taxon in question.

### 3.2 Tensor and Kronecker Products

Before discussing phylogenetic tensors, we will make a slight diversion to discuss tensor products. While scalars can be represented as zero dimensional quantities, vectors can be represented as one dimensional arrays and matrices can be represented as two dimensional arrays. Tensors are the  $n$ -dimensional generalisations of these structures. Fortunately, we can express our tensors in vector representation by using the appropriate indices. For example, suppose we wish to express a three dimensional tensor in matrix form. For a two dimensional matrix, each element must have two indices, one representing the row and one representing the column. If we were to express a three dimensional tensor in matrix form, however, each element must have three indices.

Now suppose we have two vector spaces,  $V$  and  $W$ . The tensor product of the two vector spaces is defined over some field,  $F$ , and denoted  $V \otimes W$ . It also forms a vector space.

The tensor product can be represented using the Kronecker product. For an example of the Kronecker product, suppose  $A$  is an  $m \times n$  matrix and  $B$  is a  $p \times q$  matrix. The Kronecker product of the two matrices,  $A \otimes B$ , is the  $mp \times nq$  matrix,

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{bmatrix}, \quad (3.1)$$

where  $a_{ij}B$  is the matrix  $B$ , multiplied by the scalar,  $a_{ij}$ .

### 3.3 Phylogenetic Tensors on the General Markov Model

We will start with a general discussion of phylogenetic tensors for the general Markov model. We will then derive the expressions for the two-taxon and three-taxon phylogenetic tensors for various trees and networks under the binary symmetric model.

Phylogenetic tensors are represented as column vectors of length  $m^n$ , where  $m$  and  $n$  refer to the number of states and taxa, respectively. They represent the probability distribution of the combinations of states at an arbitrary site in the character sequences and have  $m^n$  elements, each denoted  $p_{\mathbf{i}}$ . We can define a one-to-one function from the sequence alignment to the probability distribution. The probability distribution is then indexed as follows,

$$p_{\mathbf{i}} \equiv p_{i_1 i_2 \dots i_n}. \quad (3.2)$$

The index of each phylogenetic tensor element has an equivalent integer representation and  $m$ -ary representation. The function required to go from the  $m$ -ary representation to the integer

representation is

$$\mathbf{i} = 1 + i_1 m^{n-1} + i_2 m^{n-2} + \dots + i_{n-1} m + i_n = 1 + \sum_{j=1}^n i_j m^{n-j}. \quad (3.3)$$

For example, if the state space is binary ( $m = 2$ ), the subscript of the phylogenetic tensor will correspond to the combination of states with the equivalent binary representation.

We will see that this element labelling corresponds to the tensor space spanned by the tensor products of the unit vectors. For an  $m$  state Markov model, the unit vectors will be

$$e_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad e_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \dots, \quad e_{m-1} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad (3.4)$$

where each vector,  $e_j$ , has  $m$  elements, such that

$$[e_j]_i = \begin{cases} 1 & \text{if } i = j + 1, \\ 0 & \text{if } i \neq j + 1. \end{cases} \quad (3.5)$$

The phylogenetic tensor space is then spanned by the  $m^n$  dimensional space of the tensor products of the unit vectors. The tensor products of the units vectors in  $\mathbb{R}_m$  are unit vectors in  $\mathbb{R}_{m^n}$ , where the “1” entry depends on choice  $\mathbf{i}$ .

For the general Markov model, the phylogenetic tensor can now be represented as

$$P = \sum_{\mathbf{i}=1}^{m^n} p_{\mathbf{i}} e_{\mathbf{i}} \equiv \sum_{i_1, i_2, \dots, i_n=0}^{m-1} p_{i_1 i_2 \dots i_n} e_{i_1} \otimes e_{i_2} \otimes \dots \otimes e_{i_n}, \quad (3.6)$$

where  $i_k$  is the state taxon  $k$  takes at an arbitrary site in the character sequence,  $p_{i_1 i_2 \dots i_n}$  is the corresponding expression for the probability in the  $\mathbf{i} = i_1 i_2 \dots i_n$  term of the phylogenetic tensor and  $\otimes$  is the Kronecker product.

### 3.3.1 The Phylogenetic Tensor on the Binary Symmetric Model

For simplicity, the only Markov model we will examine is the binary symmetric model. For all two state models ( $m = 2$ ), such as the binary symmetric model, there are only two unit vectors to span  $\mathbb{R}_2$  and the phylogenetic tensor will be

$$P = \sum_{\mathbf{i}=1}^{2^n} p_{\mathbf{i}} e_{\mathbf{i}} \equiv \sum_{i_1, i_2, \dots, i_n=0}^1 p_{i_1 i_2 \dots i_n} e_{i_1} \otimes e_{i_2} \otimes \dots \otimes e_{i_n}. \quad (3.7)$$

### 3.4 The Splitting Operator

We are now in a position to find expressions for phylogenetic tensors in terms of the rate parameters from the Markov model and the time parameters from the tree or network structure. To find these expressions we will utilise a tool called the *splitting operator*, introduced by Bashford et al. [2004] and denoted  $\delta$ . The purpose of the splitting operator is to model the instantaneous “splitting” of one ancestral edge into two descendant edges. The splitting operator introduces a new taxon onto the tree or network. The dimension of the tensor space must therefore increase by a factor of  $m$ , the number of states in the state space, with each application of the splitting operator on the tree or network. Immediately after the splitting process has occurred, the two new taxa must have identical character sequences. In other words, the phylogenetic tensor elements representing the cases where the two new taxa are in different states must be identically zero immediately after the splitting process.

The splitting operator was first applied to phylogenetic networks by Sumner et al. [2012c]. The splitting operator maps a vector space to the tensor product space of the vector space with itself. The splitting operator is only defined on unit vectors, and maps each unit vector to the tensor product of the unit vector with itself. In summary,

$$\begin{cases} \text{i)} & \delta : V \rightarrow V \otimes V, \\ \text{ii)} & \delta \cdot e_i := e_i \otimes e_i, \\ \text{iii)} & \delta(u + v) = \delta(u) + \delta(v), \end{cases} \quad (3.8)$$

where  $V$  is a vector space,  $\otimes$  is the tensor product,  $\cdot$  is the action of the splitting operator on a vector and is represented by the matrix product,  $e_i$  is an arbitrary unit vector and  $u$  and  $v$  are arbitrary vectors in the vector space.

It is important to note that in general,

$$\delta(u) \neq u \otimes u. \quad (3.9)$$

This statement is only true when  $u \equiv e_i$ ,

$$\delta(u) = u \otimes u \Leftrightarrow u \equiv e_i. \quad (3.10)$$

For example, suppose that  $u = e_0 + e_1$ . Then

$$\delta(u) = \delta(e_0 + e_1) = \delta(e_0) + \delta(e_1) = e_0 \otimes e_0 + e_1 \otimes e_1. \quad (3.11)$$

Conversely,

$$\begin{aligned}
u \otimes u &= (e_0 + e_1) \otimes (e_0 + e_1) \\
&= e_0 \otimes e_0 + e_0 \otimes e_1 + e_1 \otimes e_0 + e_1 \otimes e_1 \\
&\neq e_0 \otimes e_0 + e_1 \otimes e_1 \\
&= \delta(e_0 + e_1) \\
&= \delta(u).
\end{aligned} \tag{3.12}$$

### 3.4.1 Pushing Back the Splitting Operator

We will now look at how the splitting operator acts on the two state general Markov model as an example. The two state general Markov model has the rate matrix,

$$Q = \begin{bmatrix} -\alpha & \beta \\ \alpha & -\beta \end{bmatrix} = \alpha L_\alpha + \beta L_\beta, \tag{3.13}$$

where columns sum to zero, the matrix element in the  $\mathbf{i}^{th}$  row and  $\mathbf{j}^{th}$  column,  $Q_{ij}$ , is the rate of substitutions from state  $\mathbf{j} - 1$  to state  $\mathbf{i} - 1$  and the generators are

$$L_\alpha = \begin{bmatrix} -1 & 0 \\ 1 & 0 \end{bmatrix}, \quad L_\beta = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}. \tag{3.14}$$

In this thesis we will be examining only the binary symmetric model, obtained by taking  $\alpha = \beta = \lambda$  in the general Markov matrix. The rate matrix for the binary symmetric model is

$$Q = \begin{bmatrix} -\lambda & \lambda \\ \lambda & -\lambda \end{bmatrix}. \tag{3.15}$$

For two state models, the matrix representation of the splitting operator is

$$\delta = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}. \tag{3.16}$$

The transition matrix,  $M$ , on an edge over time  $t$  will be the matrix exponential,  $M = e^{Qt}$ . The matrix exponential can be expressed in the form

$$M = e^{Qt} = I - \frac{1}{2} (e^{-2\lambda t} - 1) (L_\alpha + L_\beta). \tag{3.17}$$

As noted in Sumner et al. [2012c], the splitting operator performs the following action on the

transition matrix,

$$\begin{aligned}\delta \cdot M &= \delta \cdot e^{Qt} = \delta \cdot \left( I - \frac{1}{2} (e^{-2\lambda t} - 1) (L_\alpha + L_\beta) \right), \\ &= \delta \cdot I - \frac{1}{2} (e^{-2\lambda t} - 1) (\delta \cdot L_\alpha + \delta \cdot L_\beta).\end{aligned}\tag{3.18}$$

The splitting operator acts on the identity matrix and generating matrices as follows,

$$\begin{cases} \delta \cdot I &= (I \otimes I) \cdot \delta, \\ \delta \cdot L_\alpha &= (L_\alpha \otimes L_\alpha + L_\alpha \otimes I + I \otimes L_\alpha) \cdot \delta, \\ \delta \cdot L_\beta &= (L_\beta \otimes L_\beta + L_\beta \otimes I + I \otimes L_\beta) \cdot \delta. \end{cases}\tag{3.19}$$

We can check that these expressions are correct by substituting in the appropriate matrix representations for the splitting operator, the identity matrix and each of the generators.

Here it should be noted that these expressions are not unique. To see why, consider the following actions of each of the generators on a unit vector,

$$L_\beta \cdot e_0 = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},\tag{3.20}$$

$$L_\alpha \cdot e_1 = \begin{bmatrix} -1 & 0 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.\tag{3.21}$$

Now consider the following expression,

$$L_\alpha \otimes L_\beta \cdot \delta \cdot e_0 = L_\alpha \otimes L_\beta \cdot e_0 \otimes e_0 = L_\alpha \cdot e_0 \otimes L_\beta \cdot e_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.\tag{3.22}$$

Similarly,

$$L_\alpha \otimes L_\beta \cdot \delta \cdot e_1 = L_\alpha \otimes L_\beta \cdot e_1 \otimes e_1 = L_\alpha \cdot e_1 \otimes L_\beta \cdot e_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.\tag{3.23}$$

Consequently,

$$L_\alpha \otimes L_\beta \cdot \delta \cdot (e_0 + e_1) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (3.24)$$

Following similar logic,

$$L_\beta \otimes L_\alpha \cdot \delta \cdot (e_0 + e_1) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (3.25)$$

Since  $\delta$  only acts on unit vectors, we can ignore both of these terms from our expressions in a similar way to ignoring constants when performing definite integration.

Alternatively, instead of using  $L_\alpha$  and  $L_\beta$  as generators, we could have used the identity matrix and the matrix with off-diagonal entries being one and diagonal entries being zero. When we compare the action of the splitting operator on the rate matrix in both bases, we see why the extra terms that we have ignored appear. Having noted this freedom, we will not use it in this work.

Referring back to the expressions for the action of the splitting operator on the identity matrix and the generating matrices, we can see that for each expression the splitting operator is “pushed back”. Sumner et al. [2012c] showed that if we have a product of multiple transition matrices, the splitting operator can be pushed back through every transition matrix, one at a time. Eventually, the splitting operator will act on the initial probability distribution at the root.

We will now make a change of notation. Let  $\mathcal{L}_\alpha^{[2]} = L_\alpha \otimes L_\alpha + L_\alpha \otimes I + I \otimes L_\alpha$  and  $\mathcal{L}_\beta^{[2]} = L_\beta \otimes L_\beta + L_\beta \otimes I + I \otimes L_\beta$ . Also let  $R_{11} = \mathcal{L}_\alpha^{[2]} + \mathcal{L}_\beta^{[2]}$ . Later, it will become apparent why we have chosen this notation.

With this notation, the action of the splitting operator on the transition matrix is

$$\delta \cdot M = \left[ I \otimes I - \frac{1}{2} (e^{-2\lambda t} - 1) R_{11} \right] \cdot \delta. \quad (3.26)$$

From Sumner et al. [2012c] we can simplify this to

$$\begin{aligned} \delta \cdot M &= e^{\lambda R_{11} t} \cdot \delta \\ &:= \mathcal{M} \cdot \delta. \end{aligned} \quad (3.27)$$

We now have a new transition matrix and rate matrix, with the splitting operator now acting after the transition matrix.

The rate matrix for the new transition matrix is

$$\lambda R_{11} = \left[ \begin{array}{c|cccc} & 00 & 01 & 10 & 11 \\ \hline 00 & -\lambda & \lambda & \lambda & \lambda \\ 01 & 0 & -2\lambda & 0 & 0 \\ 10 & 0 & 0 & -2\lambda & 0 \\ 11 & \lambda & \lambda & \lambda & -\lambda \end{array} \right], \quad (3.28)$$

where columns sum to zero, the matrix element in the  $\mathbf{i}^{th}$  row and the  $\mathbf{j}^{th}$  column,  $[\lambda R_{11}]_{\mathbf{ij}} \equiv [\lambda R_{11}]_{i_1 i_2 j_1 j_2}$ , is the rate of substitutions from the combination of states  $j_1 j_2$  to  $i_1 i_2$ .

We now have a transition matrix that represents transitions between combinations of states for two taxa. We can see that the only substitutions that have positive rates are the substitutions that result in the two taxa having the same state.  $\lambda R_{11}$  can be used to force two edges to remain identical. The substitutions between combinations of states can be represented as a diagram, shown below in Figure 3.1.

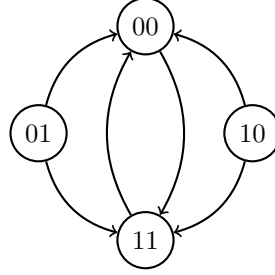


Figure 3.1:  $\lambda R_{11}$  on two taxa.

The direction of the arrows represents substitutions that can occur and each character state transition has the same rate,  $\lambda$ , from the binary symmetrical model.

### 3.4.2 The Splitting Operator on Edges

As we have seen, the splitting operator is a special function that uses the Kronecker product to increase the dimension of the phylogenetic tensor space in a way that represents the splitting of one ancestral edge into two descendant edges. Recall that the splitting operator acts directly on an individual unit vector as follows,

$$\delta \cdot e_i = e_i \otimes e_i, \quad (3.29)$$

where  $\cdot$  is the matrix product and  $\otimes$  is the Kronecker product.

The unit column vector,  $e_i$ , corresponds to the state,  $i$ , in an individual character sequence representing a single taxon. It follows that  $e_i \otimes e_i$  represents the combination of states,  $ii$ , at a given site in two character sequences representing two taxa. Immediately after a speciation



event, when no time has elapsed, the two resultant character sequences must be identical. Hence, the splitting operator follows our intuition on speciation.

Let's now look at a basic two-taxon clock-like tree with the splitting operator acting on the tree, shown in Figure 3.2 below.

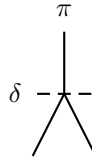


Figure 3.2: Two-taxon clock-like tree with the splitting operator,  $\delta$ , and the initial probability distribution,  $\pi$ .

The tree starts at the top, with the initial probability distribution,  $\pi$ , at the root. At the root there is only a single taxon present. The initial probability distribution is the phylogenetic tensor for the single root taxon when no time has elapsed. The number of elements in the initial probability distribution will be equal to the number of states in the state space.

We will choose the initial probability distribution to be

$$\pi = \begin{bmatrix} p_0(0) \\ p_1(0) \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}, \quad (3.30)$$

the stationary distribution. The stationary distribution is invariant under the action of any transition matrix from the chosen model on a single edge. In other words, if  $\pi$  is the stationary distribution then  $e^{Qt} \cdot \pi = \pi$  for any time,  $t$ , and for any  $Q$  from the binary symmetric model. The stationary distribution is the equilibrium distribution, which cannot be left once reached. Consequently, the stationary distribution represents the long run probability distribution for a single taxon. For an  $n$ -taxon phylogenetic tensor, the initial probability distribution can be recovered from the marginal probability distributions for each taxon.

The vertical line down from the root is the edge representing the evolution of the single taxon over time from the root. Eventually this single taxon splits into two new descendant taxa. The dotted line represents the stage in time where the splitting operator is used to represent the splitting of the single ancestral taxon into two descendant taxa. Over time the two new taxa diverge from each other, represented by two edges which move further apart through time.

After placing the splitting operator wherever a splitting event occurs on a tree, we can push each splitting operator through the tree. After pushing the splitting operators through edges towards the root, it is equivalent to think of a single edge as a collection of edges being forced to remain identical, with the number of edges equal to the number of descendant taxa. When one splitting operator is pushed back, two edges become identical. The number of identical edges is equal to one plus the number of splitting operators below it which have been pushed back above the edge. In Figure 3.3 below is an example of the action the splitting operator performs on a single taxon, transforming it into a two-taxon tree.

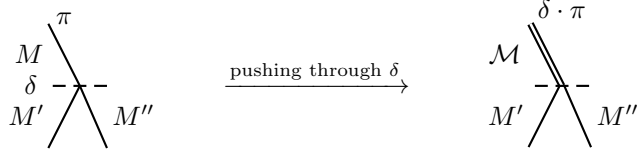


Figure 3.3: Action of the splitting operator on a single taxon, transforming it into a two-taxon tree.  $M$ ,  $M'$ ,  $M''$  and  $\mathcal{M}$  refer to the Markov matrices on those edges.  $\pi$  is the probability distribution at the root.

Algebraically, the above scenario is given by  $(M' \otimes M'') \cdot (\delta \cdot M) \cdot \pi \rightarrow (M' \otimes M'') \cdot \mathcal{M} \cdot \delta \cdot \pi$ , where the edge labels are the transition matrices for the edges in question.

The consequence is that the splitting operator now acts further up on the tree or network, in this case at the root. If the root is further up the tree or network, we can repeat this process as many times as necessary until the splitting operator acts directly on the root. Likewise, we can perform this process on every splitting operator on the tree or network. Every splitting operator would then act directly on the root.

$\mathcal{M}$  plays the role of implementing correlated changes. By pushing back the splitting operator, we now have a rate matrix which models multiple identical edges being forced to remain identical. An intriguing question that arises from this result is “what happens when this rate matrix is applied to multiple adjacent non-identical edges?”. Sumner et al. [2012c] discussed the issue of whether the rate matrices arising from pushing back the splitting operator can be used to model convergence on parts of a network which are isolated from splitting processes.

### 3.5 Modelling Convergence

Recall the rate matrix that arises when we push back the splitting operator,

$$\lambda R_{11} = \left[ \begin{array}{c|cccc} & 00 & 01 & 10 & 11 \\ \hline 00 & -\lambda & \lambda & \lambda & \lambda \\ 01 & 0 & -2\lambda & 0 & 0 \\ 10 & 0 & 0 & -2\lambda & 0 \\ 11 & \lambda & \lambda & \lambda & -\lambda \end{array} \right]. \quad (3.31)$$

We have seen that the rate matrix formed by pushing back the splitting operator can be used to model identical edges. We will argue that it can also be used to model the convergence of two edges in the same time period which have previously diverged from each other. We can simply model the convergence of these two diverged edges with a single transition matrix and the above rate matrix. For the binary symmetric model, the rates of substitutions leading to identical states will be the same as the rates of substitutions on individual edges. Rates of substitutions away from identical states will be zero.

We can generalise this concept to model the convergence of any subset of edges with each other in a given time period. We are not restricted to one group of converging edges. We can

have multiple groups of converging edges in one time period. We will also see that we can have groups of more than two taxa converging with each other in a given time period.

We will represent converging edges graphically with curved lines. Below in Figure 3.4 is a simple example of a two-taxon network with two converging edges.

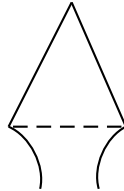


Figure 3.4: Two-taxon clock-like convergence-divergence network with a convergence period in the second time period.

On our trees and networks, divergence is represented by straight lines emanating from a node, while convergence is represented by curved lines. The root is at the top of the tree or network and time progresses linearly down the page. The other dimension in the two-dimensional plane on the page has no meaning other than to create space for the diagram. By moving across the page we are not moving through time or any other dimension, we are simply moving from one edge to another. Consequently, we can reflect the edges below any node in any tree or network. In this thesis the labelling of each tree and network should be considered independently unless otherwise stated. Time parameters and leaf labels should not be equated or compared between one tree or network and another unless explicitly stated.

### 3.6 Constructing the Trees and Networks

Phylogenetic tensors can be expressed in many algebraically equivalent forms. It is perhaps most convenient conceptually to express the phylogenetic tensor as a series of matrix products, with each matrix being a transition matrix representing a distinct, non-overlapping period in time. We first partition the tree or network into these time periods, with time periods being separated by splitting events or the start or end of convergence periods. We can even express unrooted trees this way by choosing an arbitrary placement of the root. Each transition matrix is a matrix exponential, with its argument being the sum of the evolutionary processes happening in that time period.

We will now introduce some notation to keep track of which taxa are converging or diverging in different time periods. Let  $[n]$  denote the set of all taxa and  $A_1^k = i_1 i_2 \dots i_n \subseteq [n]$  be a subset of edges for a convergence group in time period  $k$ , represented by an ordered binary string concatenation of length  $n$ , with “1” representing a taxon in the group and “0” representing a taxon not in the group.

Since they are algebraically equivalent, there is no distinction made between a group of edges undergoing convergence and a single diverging edge “converging” with itself. A single diverging

edge can be treated as its own convergence group. We will restrict each edge to belong to exactly one convergence group. Consequently,  $A_i^k \cap A_j^k = \emptyset$  for all  $i \neq j$  and  $\bigcup A_i^k = A_1^k \cup A_2^k \cup \dots \cup A_l^k = [n]$ . Below in Figure 3.5 is an example of the notation on a three-taxon tree.

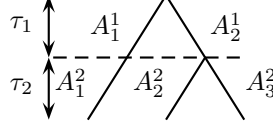


Figure 3.5: Three-taxon clock-like tree.  $A_1^1 = A_1^2 = 100$ ,  $A_2^1 = 011$ ,  $A_2^2 = 010$  and  $A_3^2 = 001$ .

The rate matrix for convergence of edges,  $A_i^k$ , is expressed as

$$R_{A_i^k} = \lambda \sum_{B \subseteq A_i^k, B \neq \emptyset} (R_{B,\alpha} + R_{B,\beta}), \quad (3.32)$$

where  $R_{B,x} = S_{i_1,x} \otimes S_{i_2,x} \otimes \dots \otimes S_{i_n,x}$  and  $S_{i_r,x} = \begin{cases} I & \text{if } i_r = 0, \\ L_x & \text{if } i_r = 1. \end{cases}$

As a further example, suppose  $n = 3$ . The rate matrix for the convergence of the first two taxa, with the third taxa diverging, is then

$$\lambda R_{110} = \lambda (L_\alpha \otimes L_\alpha \otimes I + L_\alpha \otimes I \otimes I + I \otimes L_\alpha \otimes I + L_\beta \otimes L_\beta \otimes I + L_\beta \otimes I \otimes I + I \otimes L_\beta \otimes I)$$

$$= \begin{bmatrix} & 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\ \begin{matrix} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{matrix} & \begin{bmatrix} -\lambda & 0 & \lambda & 0 & \lambda & 0 & \lambda & 0 \\ 0 & -\lambda & 0 & \lambda & 0 & \lambda & 0 & \lambda \\ 0 & 0 & -2\lambda & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2\lambda & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2\lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2\lambda & 0 & 0 \\ \lambda & 0 & \lambda & 0 & \lambda & 0 & -\lambda & 0 \\ 0 & \lambda & 0 & \lambda & 0 & \lambda & 0 & -\lambda \end{bmatrix} \end{bmatrix}. \quad (3.33)$$

We will set  $\lambda t_i = \tau_i$  for convenience since there is only one degree of freedom in the product  $\lambda t_i$ . Finally, the general phylogenetic tensor for  $n$  taxa can be expressed as

$$P = \left( \prod_{k=1}^m e^{\left( \sum_i R_{A_i^{m+1-k}} \right) \tau_{m+1-k}} \right) \cdot \Pi, \quad (3.34)$$

where  $m$  is the number of epochs and the summation is over all convergence groups in the time period in question and  $\Pi = \Delta \cdot \pi$ . We have let  $\Pi$  denote the initial probability distribution after all splitting operators have been pushed back, with  $\Delta$  being the splitting operator that maps

the initial probability distribution on the root taxon,  $\pi$ , to the initial probability distribution on all  $n$  taxa,  $\Pi$ . It should be noted that the product counts backwards, as  $k = 1$  gives the time parameter  $\tau_m$  and  $k = m$  gives the time parameter  $\tau_1$ . This is because the matrix exponentials start with the most recent time period, the  $m^{th}$  time period, on the left and end with the first time period on the right.

Now we will look at a three-taxon convergence-divergence network in Figure 3.6 below as an example.

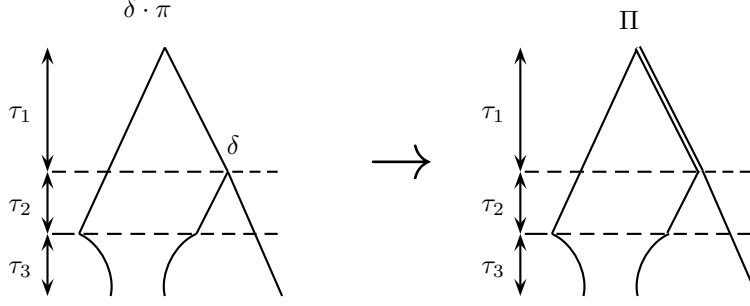


Figure 3.6: A three-taxon convergence-divergence network.

The phylogenetic tensor for this network can be written as

$$P = e^{(R_{110}+R_{001})\tau_3} \cdot e^{(R_{100}+R_{010}+R_{001})\tau_2} \cdot e^{(R_{100}+R_{011})\tau_1} \cdot \Pi. \quad (3.35)$$

As an example, the convergence of the first and second taxa, in conjunction with the divergence of the third taxon, in the third time interval, is described by the rate matrix,

$$\lambda(R_{110} + R_{001}) = \begin{bmatrix} & 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\ \begin{matrix} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{matrix} & \begin{bmatrix} -2\lambda & \lambda & \lambda & 0 & \lambda & 0 & \lambda & 0 \\ \lambda & -2\lambda & 0 & \lambda & 0 & \lambda & 0 & \lambda \\ 0 & 0 & -3\lambda & \lambda & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda & -3\lambda & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -3\lambda & \lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda & -3\lambda & 0 & 0 \\ \lambda & 0 & \lambda & 0 & \lambda & 0 & -2\lambda & \lambda \\ 0 & \lambda & 0 & \lambda & 0 & \lambda & \lambda & -2\lambda \end{bmatrix} \end{bmatrix}. \quad (3.36)$$

Note that this rate matrix is the sum of the rate matrix for the convergence of the first two taxa, which we gave as an example earlier, and the rate matrix for the divergence of the third taxon from the other two taxa.

The substitutions are represented diagrammatically in Figure 3.7 below.

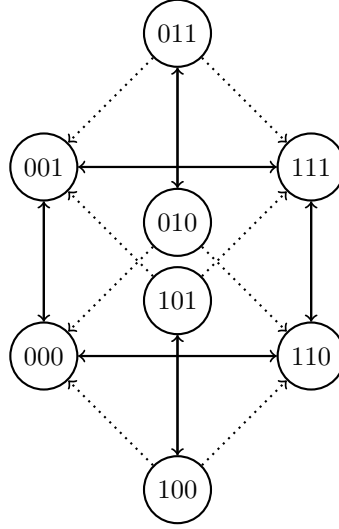


Figure 3.7:  $\lambda(R_{110} + R_{001})$  on three edges. Each transition has the same rate,  $\lambda$ , from the binary symmetrical model. Solid lines indicate “regular” mutations, while dotted lines indicate “correction” mutations, responsible for convergence.

### 3.7 Transformed Phylogenetic Tensors in the Hadamard Basis

We transform the basis of the  $n$ -taxon phylogenetic tensor by multiplying it on the left by the transforming matrix,  $H = h \otimes h \otimes \dots \otimes h$ , where there are  $n - 1$  tensor products. Matrices which are in the transformed basis will be denoted by a “ $\hat{\phantom{x}}$ ” above them. We showed in Chapter 2 that the general transforming matrix,  $h$ , for the binary symmetric model is

$$h = \begin{bmatrix} k_1 & k_2 \\ k_1 & -k_2 \end{bmatrix}. \quad (3.37)$$

If we let  $k_1 = k_2 = 1$ , the first row of  $H$  multiplied by the phylogenetic tensor will give the sum of all of the probability distribution elements and must be equal to one due to probability conservation. By letting  $k_1 = k_2 = 1$ ,  $h$  becomes the Hadamard matrix,

$$h = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (3.38)$$

which diagonalises  $Q$  to give

$$\hat{Q} = h \cdot Q \cdot h^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & -2\lambda \end{bmatrix}. \quad (3.39)$$

It transforms the Markov generators to

$$\widehat{L}_\alpha = h \cdot L_\alpha \cdot h^{-1} = \begin{bmatrix} 0 & 0 \\ -1 & -1 \end{bmatrix}, \quad \widehat{L}_\beta = h \cdot L_\beta \cdot h^{-1} = \begin{bmatrix} 0 & 0 \\ 1 & -1 \end{bmatrix}. \quad (3.40)$$

The initial probability distribution becomes  $\widehat{\Pi} = H \cdot \Delta \cdot \pi$ .

Finally, the transformed basis of the general phylogenetic tensor will be

$$\widehat{P} = H \cdot P = \left( \prod_{k=1}^m e^{\left( \sum_i \widehat{R}_{A_i^{m+1-k}} \right) \tau_{m+1-k}} \right) \cdot \widehat{\Pi}. \quad (3.41)$$

Applying the Hadamard transformation results in  $\widehat{p}_{00\dots 0} = q_{00\dots 0} = 1$ , the conservation of probability. All other elements of  $\widehat{P}$  will be linear combinations of the elements of  $P$ , with coefficients of  $\pm 1$ , the elements of the Hadamard matrix.

Since there is only a single degree of freedom in  $\tau_i = \lambda t_i$  we are free to rescale  $\tau_i \rightarrow \frac{1}{2}\tau_i$  to simplify our expressions.

### 3.8 Modelling Convergence on Networks

Before we proceed to examine some two-taxon, three-taxon and four-taxon trees and networks, we will first establish some criteria for determining permissible convergence-divergence networks.

Our convergence-divergence networks will be partitioned into distinct non-overlapping time periods. We will call a set of edges converging together in an time period a *convergence group*. A new time period starts directly after a splitting event or when the convergence groups change. For example, there may be a splitting event giving rise to a new time period where all edges diverge from each other. A group of edges may then converge with each other in a new time period. Finally, the convergence group(s) may change, giving rise to another new time period. All time periods containing convergence groups must occur immediately after a divergence period or another convergence period. For diagrammatic examples of converge-divergence networks see Figure 3.6 on page 63 or Figure 3.8 on page 67.

Suppose we have an  $n$ -taxon clock-like tree.  $n$ -taxon clock-like trees will have  $n - 1$  numerical parameters. If we wish to adopt a more parameter rich tree we can drop the clock-like assumption. Non-clock-like trees will have  $2n - 3$  numerical parameters. For  $n \geq 3$ ,  $2n - 3 > n - 1$  and non-clock-like trees will be more parameter-rich than clock-like trees.  $2n - 3 - (n - 1) = n - 2$  and for large  $n$  non-clock-like trees will have approximately double the number of numerical parameters that clock-like trees with the same number of taxa have. In contrast, convergence-divergence networks introduce any positive integer number of extra parameters to the clock-like tree. A clock-like tree may not be parameter rich enough to be a “good” fit for a set of pattern frequencies and a non-clock-like tree may overfit the pattern frequencies by providing too many parameters. In this case a convergence-divergence network could be used to introduce a number

of extra parameters to the clock-like tree which is less than the number of extra parameters introduced by removing the clock assumption.

We could potentially have an unlimited number of time periods, each with different convergence groups to the time period immediately prior to it. Increasing the number of parameters raises the question of potential overfitting, however. To avoid networks that may overfit or make the phylogenetic tensor expressions too complicated, we will only look at examples with relatively few time periods.

We have seen that if a clock-like tree does not have enough parameters to fit a particular set of pattern frequencies then we can introduce more parameters by either removing the clock-like assumption or introducing time periods with convergence. We also have the option of a non-clock-like convergence-divergence network, where we both remove the clock-like assumption and introduce time periods with convergence. A non-clock-like convergence-divergence network may be preferred when no direction of time is known or there is rate heterogeneity across edges in a given time period and there is reason to believe that the assumption of independent evolution has broken down at some stage in time, leading to convergence of edges. For simplicity, we will only examine clock-like convergence-divergence networks however.

It is necessary to place some restrictions on convergence periods so that the convergence-divergence networks to examine are manageable. We will restrict the convergence periods to allow for the more biologically realistic convergence-divergence networks. For example, without any restrictions on convergence periods we could have two edges initially diverging from a node, followed by the same two edges alternating between converging and diverging indefinitely. Another potential issue is having too many allowable groups of converging edges in a time period. The more edges there are across a time period on a network, the more potential convergence groups there could be among these edges. We will place some restrictions on which groups of edges can convergence on a network.

For an  $n$ -taxon tree or network and a Markov model with  $m$  states, there will be  $m^n$  elements in the phylogenetic tensor. If there are greater than  $m^n$  numerical parameters, then there is no possibility of the tree or network being identifiable. We will therefore restrict the number of convergence periods on our networks accordingly.

We will now introduce some restrictions on our convergence time periods. These restrictions will be:

1. In each time period, each edge can only be included in one convergence group.
2. There will be a maximum of one convergence period between each splitting event and after the last splitting event.

Any subset of edges, 2, 3,  $\dots$ ,  $w$ , where  $w$  is the total number of edges at that time period, after  $w - 1$  splitting events, can converge together. An individual edge can also be thought of as “converging” with itself when it is diverging from all other edges in the time period, as described earlier in Chapter 3.



Now consider a three-taxon clock-like tree. This tree has two splitting events, the first represented by the root and the second by a node which splits one of the diverged edges into two new edges. The first place where we can put a convergence period is before the second splitting event, which will represent convergence of the two diverged edges. The second place a convergence period can be located is after the second node and after divergence of the third edge has occurred. We can represent convergence of any two edges or of all three edges. Below in Figure 3.8 is an example of a three-taxon network with a convergence period after the divergence of the first two edges and a convergence period involving the non-sister taxa after a third edge has split off.

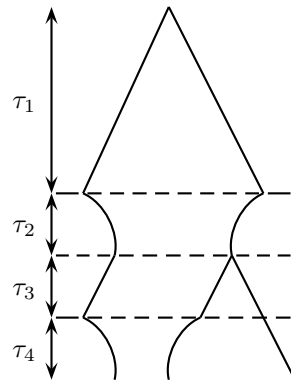


Figure 3.8: An example of a three-taxon clock-like network. The two edges are converging in the second time period and the non-sister taxa are converging in the fourth time period.



## Chapter 4

# Two-Taxon and Three-Taxon Trees and Networks

In Chapter 3 we laid out the necessary notation to express phylogenetic tensors for trees and networks in the Hadamard basis. In this chapter we will present the phylogenetic tensors for some basic two-taxon and three-taxon trees and networks. We will address the question of identifiability. That is, if the tree or network structure is fixed, is there a bijective function between the parameter space and the pattern frequencies? The constraints on each phylogenetic tensor will provide a convenient basis for determining the theoretical character sequence patterns that can arise from that tree or network. In this chapter we will compare the constraints and the probability spaces that arise from the constraints of various trees and networks under the binary symmetric model and determine which trees and networks are distinguishable from each other.

Recall from Chapter 3 that rates of substitutions cannot be distinguished from time along an edge for the binary symmetric model. In other words, we can only ever know the product of the rate and the time. This product can be large from either a fast rate or a long period of time. Likewise, the product can be small from either a slow rate or a short period time. We are therefore able to rescale our rate and time parameters into a single parameter. Recall in Section 3.7 that we rescaled  $\tau_i = \lambda t_i$  by letting  $\tau_i \rightarrow \frac{1}{2}\tau_i$  since there is only a single degree of freedom in  $\tau_i = \lambda t_i$ . This rescaling will allow our expressions for the transformed phylogenetic tensor elements to be compared more easily to distances between leaves on the tree. For example, with this rescaling one element of the transformed phylogenetic tensor for the two-taxon clock-like tree is  $q_{11} = e^{-2\tau_1}$ , where  $2\tau_1$  is the sum of the edge lengths between the two taxa.

### 4.1 Constraints on the Transformed Phylogenetic Tensors

Before we introduce the two-taxon and three-taxon trees and networks, we will have a look at how the Hadamard transformation impacts on the phylogenetic tensors under the binary symmetric

model.

The phylogenetic tensors are represented as  $2^n$  dimensional vectors, where there are 2 states in the Markov model and  $n$  taxa on the tree or network. The tensor elements are ordered according to the base-2 numeral system, with each element in the form,

$$p_{i_1 i_2 \dots i_n}, \quad (4.1)$$

where  $i_1, i_2, \dots, i_n \in \{0, 1\}$ .

In the transformed basis, the phylogenetic tensor will be expressed in vector representation as the product of the diagonalising matrix,  $h^{\otimes n}$ , and the vector representation of the phylogenetic tensor in the untransformed basis. Recall that we expressed the transformed phylogenetic tensor as

$$\hat{P} = H \cdot P = h^{\otimes n} \cdot P. \quad (4.2)$$

Recall from Chapter 2 that if the Markov model is the binary symmetric model then the diagonalising matrix,  $h$ , will be

$$h = \begin{bmatrix} k_1 & k_2 \\ k_1 & -k_2 \end{bmatrix}, \quad (4.3)$$

where  $k_1, k_2 \in \mathbb{C} \setminus \{0\}$ . If we choose  $k_1 = k_2 = 1$ , the diagonalising matrix becomes a Hadamard matrix.

For an  $n$ -taxon tree or network under the binary symmetric model, the diagonalising matrix will be

$$H = h^{\otimes n} = h \otimes h \otimes \dots \otimes h, \quad (4.4)$$

where there are  $n - 1$  tensor products.

If we index rows and columns from 0 upwards then the elements of the Hadamard matrix of order 2 are

$$h_{\mathbf{ij}} \equiv h_{i_1 i_2, j_1 j_2} = (-1)^{i_1 j_1 + i_2 j_2}. \quad (4.5)$$

The elements of the Hadamard matrix of order  $n$  will be

$$H_{\mathbf{ij}} \equiv H_{i_1 i_2 \dots i_n, j_1 j_2 \dots j_n} = (-1)^{\sum_{l=1}^n i_l j_l}, \quad (4.6)$$

since the elements with the appropriate indices are simply multiplied together under the tensor product.

Since Hadamard matrices are orthogonal matrices and our choice of  $k_1 = k_2 = 1$  resulted in

every element of the first row being equal to 1, every other row must have exactly half of its elements being 1 and the other half being  $-1$ . For the binary symmetric model, the functions for the phylogenetic tensor elements come in identical pairs in the untransformed basis. If every 0 is replaced with a 1 and every 1 is replaced by a 0 the resulting phylogenetic tensor element will be identical to the original phylogenetic tensor element. We express this as

$$p_{i_1 i_2 \dots i_n} = p_{\bar{i}_1 \bar{i}_2 \dots \bar{i}_n}, \quad (4.7)$$

where

$$\begin{cases} \bar{i}_l = 0 & \text{if } i_l = 1 \\ \bar{i}_l = 1 & \text{if } i_l = 0 \end{cases}, \quad (4.8)$$

or  $\bar{i}_l = \frac{1}{2} (1 + (-1)^{i_l})$ .

The elements of the transformed phylogenetic tensor will be

$$\begin{aligned} q_{i_1 i_2 \dots i_n} &= \sum_{j_1, j_2, \dots, j_n=0}^1 H_{i_1 i_2 \dots i_n, j_1 j_2 \dots j_n} p_{j_1 j_2 \dots j_n} \\ &= \sum_{j_1, j_2, \dots, j_n=0}^1 (-1)^{\sum_{l=1}^n i_l j_l} p_{j_1 j_2 \dots j_n}. \end{aligned} \quad (4.9)$$

Note that the first element of the transformed phylogenetic tensor will represent the conservation of probability,

$$q_{00 \dots 0} = \sum_{i_1, i_2, \dots, i_n=0}^1 p_{i_1 i_2 \dots i_n} = 1. \quad (4.10)$$

Suppose the only information about a phylogenetic tensor that is known is that each element,  $p_{i_1 i_2 \dots i_n}$ , is a probability. Then

$$0 \leq p_{i_1 i_2 \dots i_n} \leq 1 \quad (4.11)$$

for all  $i_1, i_2, \dots, i_n = 0, 1$ , and

$$\sum_{i_1, i_2, \dots, i_n=0}^1 p_{i_1 i_2 \dots i_n} = 1. \quad (4.12)$$

Since every element of the Hadamard matrix is  $\pm 1$ , every element of the transformed phylogenetic tensor must be equal to the sum of a subset of elements of the untransformed phylogenetic tensor minus the remaining elements of the untransformed phylogenetic tensor elements. We can

express this as

$$\begin{aligned}
 q_{i_1 i_2 \dots i_n} &= \sum_A p_{i_1 i_2 \dots i_n} - \sum_{A^C} p_{i_1 i_2 \dots i_n} \\
 &= \sum_A p_{i_1 i_2 \dots i_n} - \left( 1 - \sum_A p_{i_1 i_2 \dots i_n} \right) \\
 &= 2 \sum_A p_{i_1 i_2 \dots i_n} - 1,
 \end{aligned} \tag{4.13}$$

where  $A$  is some subset of the indices of the phylogenetic tensor elements and  $A^C$  is the complementary set and represents all of the remaining indices of the phylogenetic tensor elements. If

$$\sum_A p_{i_1 i_2 \dots i_n} = 1, \tag{4.14}$$

then

$$q_{i_1 i_2 \dots i_n} = 1. \tag{4.15}$$

If

$$\sum_A p_{i_1 i_2 \dots i_n} = 0, \tag{4.16}$$

then

$$q_{i_1 i_2 \dots i_n} = -1. \tag{4.17}$$

We can conclude that in general, with the Hadamard matrix as the diagonalising matrix, the transformed phylogenetic tensor elements will satisfy

$$-1 \leq q_{i_1 i_2 \dots i_n} \leq 1, \tag{4.18}$$

for all  $i_1, i_2, \dots, i_n = 0, 1$ .

Now recall that the elements of the transformed phylogenetic tensor will be

$$q_{i_1 i_2 \dots i_n} = \sum_{j_1, j_2, \dots, j_n=0}^1 (-1)^{\sum_{l=1}^n i_l j_l} p_{j_1 j_2 \dots j_n}. \tag{4.19}$$

Only if  $i_l = 1$  will  $j_l$  appear in the sum in the exponent. For example,

$$\begin{aligned} q_{01101} &= \sum_{j_1, j_2, \dots, j_5=0}^1 (-1)^{\sum_{l=1}^5 i_l j_l} p_{j_1 j_2 \dots j_5} \\ &= \sum_{j_1, j_2, \dots, j_5=0}^1 (-1)^{j_2 + j_3 + j_5} p_{j_1 j_2 \dots j_5}. \end{aligned} \quad (4.20)$$

If  $\sum_{l=1}^n i_l$  is odd then there will be an odd number of terms in the exponent,  $\sum_{l=1}^n i_l j_l$ . If the sum of the exponent terms is odd then  $(-1)^{\sum_{l=1}^n i_l j_l}$ , the coefficient for  $p_{j_1 j_2 \dots j_n}$ , will be negative. The sum of the complement combination,  $\sum_{l=1}^n i_l \bar{j}_l$ , will then be even and the coefficient for  $p_{\bar{j}_1 \bar{j}_2 \dots \bar{j}_n}$ ,  $(-1)^{\sum_{l=1}^n i_l \bar{j}_l}$ , will be positive. Hence, if  $\sum_{l=1}^n i_l$  is odd then  $q_{i_1 i_2 \dots i_n} = 0$  for all trees and networks under the binary symmetric model.

If  $\sum_{l=1}^n i_l$  is even then there will be an even number of terms in the exponent,  $\sum_{l=1}^n i_l j_l$ . Again, if the sum of the exponent terms is odd then  $(-1)^{\sum_{l=1}^n i_l j_l}$ , the coefficient for  $p_{j_1 j_2 \dots j_n}$ , will be negative. However, the sum of the complement combination,  $\sum_{l=1}^n i_l \bar{j}_l$ , will be odd too and  $(-1)^{\sum_{l=1}^n i_l \bar{j}_l}$ , the coefficient for  $p_{\bar{j}_1 \bar{j}_2 \dots \bar{j}_n}$ , will also be negative. Likewise, if the coefficient for  $p_{j_1 j_2 \dots j_n}$  is positive then the coefficient for  $p_{\bar{j}_1 \bar{j}_2 \dots \bar{j}_n}$  will also be positive. Hence, we cannot assume under the binary symmetric model with no tree or network structure that  $q_{i_1 i_2 \dots i_n} = 0$  if  $\sum_{l=1}^n i_l$  is even.

For example, we will again look at  $q_{01101}$ . For every combination of  $j_2 j_3 j_5$  for which  $j_2 + j_3 + j_5$  is even and  $(-1)^{j_2 + j_3 + j_5} = 1$ , the complement combination,  $\bar{j}_2 \bar{j}_3 \bar{j}_5$ , will have  $\bar{j}_2 + \bar{j}_3 + \bar{j}_5$  being odd and  $(-1)^{\bar{j}_2 + \bar{j}_3 + \bar{j}_5} = -1$ . This means that the coefficients for  $p_{i_1 i_2 \dots i_5}$  and  $p_{\bar{i}_1 \bar{i}_2 \dots \bar{i}_5}$  must always have the opposite sign since we are summing over all possible values of  $j_1$  and  $j_4$  as well. Hence, we must have  $q_{01101} = 0$ .

As a second example, we will look at  $q_{01100}$ . For every combination of  $j_2 j_3$  for which  $j_2 + j_3$  is odd and  $(-1)^{j_2 + j_3} = -1$ , the complement combination,  $\bar{j}_2 \bar{j}_3$ , will have  $\bar{j}_2 + \bar{j}_3$  being odd and  $(-1)^{\bar{j}_2 + \bar{j}_3} = -1$ . This means that the coefficients for  $p_{i_1 i_2 \dots i_5}$  and  $p_{\bar{i}_1 \bar{i}_2 \dots \bar{i}_5}$  must both be negative. Likewise, for every combination of  $j_2 j_3$  for which  $j_2 + j_3$  is even and  $(-1)^{j_2 + j_3} = 1$ , the complement combination,  $\bar{j}_2 \bar{j}_3$ , will have  $\bar{j}_2 + \bar{j}_3$  being even and  $(-1)^{\bar{j}_2 + \bar{j}_3} = 1$ . This means that the coefficients for  $p_{i_1 i_2 \dots i_5}$  and  $p_{\bar{i}_1 \bar{i}_2 \dots \bar{i}_5}$  must both be positive. This means that the coefficients for  $p_{i_1 i_2 \dots i_5}$  and  $p_{\bar{i}_1 \bar{i}_2 \dots \bar{i}_5}$  must always have the same sign since we are summing over all possible values of  $j_1$ ,  $j_4$  and  $j_5$  as well.

## 4.2 Two-Taxon Phylogenetic Tensors

We will start by presenting expressions for the phylogenetic tensors for two-taxon trees and networks. There are two trees, the non-clock-like tree and the clock-like tree, and one network, the clock-like convergence-divergence network, that we will examine. The two-taxon clock-like convergence-divergence network is simply the two-taxon clock-like tree, with the two taxa converging with each other in the last time period.

### 4.2.1 Non-Clock-Like Tree

The two-taxon non-clock-like tree is shown below in Figure 4.1.

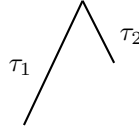


Figure 4.1: Two-taxon non-clock-like tree.

The phylogenetic tensor for this tree is

$$\hat{P} = e^{\frac{1}{2}(\hat{R}_{10}\tau_1 + \hat{R}_{01}\tau_2)} \cdot \hat{\Pi} = \begin{bmatrix} q_{00} \\ q_{01} \\ q_{10} \\ q_{11} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ e^{-(\tau_1 + \tau_2)} \end{bmatrix}. \quad (4.21)$$

Clearly the numerical parameters of this tree are not recoverable and the tree is not identifiable since there are two numerical parameters and only one variable phylogenetic tensor element. For each choice of the non-negative numerical parameters there will be a single phylogenetic tensor. The converse is not true, however. For every phylogenetic tensor there will be infinitely many numerical parameter combinations that can give rise to that tensor. It is only the sum of the two numerical parameters that the phylogenetic tensor is dependent on. There are infinitely many choices of the two numerical parameters that will result in the same sum and hence the same phylogenetic tensor.

Demanding  $\tau_1, \tau_2 \geq 0$ , the minimum length interval containing all possible values of the one variable phylogenetic tensor element is

$$0 < q_{11} = e^{-(\tau_1 + \tau_2)} \leq 1. \quad (4.22)$$

In summary, the constraints on the phylogenetic tensor for the two-taxon non-clock-like tree



are

$$\left\{ \begin{array}{l} q_{00} = 1, \\ q_{01} = 0, \\ q_{10} = 0, \\ 0 < q_{11} \leq 1. \end{array} \right. \quad (4.23)$$

We will now look at the remaining two two-taxon trees and networks and compare the phylogenetic tensor constraints.

### 4.2.2 Clock-Like Tree

The next tree we will consider is the two-taxon clock-like tree, shown in Figure 4.2 below.



Figure 4.2: Two-taxon clock-like tree.

The phylogenetic tensor for this tree is

$$\hat{P} = e^{\frac{1}{2}(\hat{R}_{10} + \hat{R}_{01})\tau_1} \cdot \hat{\Pi} = \begin{bmatrix} q_{00} \\ q_{01} \\ q_{10} \\ q_{11} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ e^{-2\tau_1} \end{bmatrix}. \quad (4.24)$$

We can see that the phylogenetic tensor for the clock-like tree can be recovered from the phylogenetic tensor for the non-clock-like tree by simply forcing the two time parameters to be equal. The phylogenetic tensor elements for the two trees are very similar, with the only difference being the argument of the exponential in the single variable element. For the clock-like tree, there is only one numerical parameter and only one variable phylogenetic tensor element. The exponential function maps the single non-positive variable to the interval  $(0, 1]$  and is a bijective function. Therefore, the two-taxon clock-like tree must be identifiable.

As with the non-clock-like tree, the constraints on the phylogenetic tensor for the two-taxon clock-like tree are

$$\left\{ \begin{array}{l} q_{00} = 1, \\ q_{01} = 0, \\ q_{10} = 0, \\ 0 < q_{11} \leq 1. \end{array} \right. \quad (4.25)$$

### 4.2.3 Clock-Like Convergence-Divergence Network

The final two-taxon tree or network is the two-taxon convergence-divergence network, shown below in Figure 4.3. This is the two-taxon clock-like tree with convergence of the two taxa immediately after the divergence.

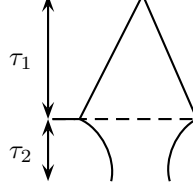


Figure 4.3: Two-taxon clock-like convergence-divergence network with a convergence period in the second time period.

The phylogenetic tensor for this network is

$$\hat{P} = e^{\frac{1}{2}\hat{R}_{11}\tau_2} \cdot e^{\frac{1}{2}(\hat{R}_{10} + \hat{R}_{01})\tau_1} \cdot \hat{\Pi} = e^{\frac{1}{2}\hat{R}_{11}\tau_2} \cdot \hat{P}_{cl} = \begin{bmatrix} q_{00} \\ q_{01} \\ q_{10} \\ q_{11} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 - e^{-\tau_2} (1 - e^{-2\tau_1}) \end{bmatrix}. \quad (4.26)$$

Clearly this network is not identifiable since there are two numerical parameters and only one variable phylogenetic tensor element and hence the numerical parameters cannot be recovered independently.

Unlike the previous two trees, it is not immediately clear what the minimum length interval that contains all of the possible phylogenetic tensors is. Demanding  $\tau_1 \geq 0$ ,

$$\begin{aligned} 0 &< e^{-2\tau_1} \leq 1 \\ \Leftrightarrow 0 &\leq 1 - e^{-2\tau_1} < 1. \end{aligned} \quad (4.27)$$

Also, demanding  $\tau_2 \geq 0$ ,

$$0 < e^{-\tau_2} \leq 1. \quad (4.28)$$

The product of these two components must also lie between 0 and 1, with the interval including 0, but not including 1.

$$\begin{aligned} 0 &\leq e^{-\tau_2} (1 - e^{-2\tau_1}) < 1 \\ \Leftrightarrow 0 &< 1 - e^{-\tau_2} (1 - e^{-2\tau_1}) \leq 1. \end{aligned} \quad (4.29)$$

As with the other two trees, the constraints on the network are

$$\begin{cases} q_{00} = 1, \\ q_{01} = 0, \\ q_{10} = 0, \\ 0 < q_{11} \leq 1. \end{cases} \quad (4.30)$$

#### 4.2.4 Network Identifiability

We established in the previous sections that while the clock-like tree is identifiable and the convergence-divergence network is not, both the clock-like tree and the convergence-divergence network have the same phylogenetic tensor constraints. This raises the question of whether the convergence-divergence network can be compared to the clock-like tree with less divergence. For example, suppose we have the convergence-divergence network with divergence over time  $\tau'_1$ , followed by convergence over time  $\tau'_2$ . Is this network identifiable from a clock-like tree with divergence over time  $\tau_1 \leq \tau'_1$ ?

Now suppose we have the phylogenetic tensor element,  $q_{11}$ , and wish to compare the time variables on the clock-like tree with no convergence to the clock-like convergence-divergence network. If we assume  $0 < q_{11} \leq 1$  such that it could have arisen on either tree or network, we require

$$\begin{aligned} q_{11} &= e^{-2\tau_1} = 1 - e^{-\tau'_2} (1 - e^{-2\tau'_1}) \\ \Leftrightarrow e^{-2\tau_1} - 1 &= -e^{-\tau'_2} (1 - e^{-2\tau'_1}) \\ \Leftrightarrow 1 - e^{-2\tau_1} &= e^{-\tau'_2} (1 - e^{-2\tau'_1}). \end{aligned} \quad (4.31)$$

Since  $0 < 1 - e^{-2\tau_1} \leq 1$ ,  $0 < e^{-\tau'_2} \leq 1$  and  $0 < 1 - e^{-2\tau'_1} \leq 1$ , we can conclude that

$$\begin{aligned} 1 - e^{-2\tau_1} &\leq 1 - e^{-2\tau'_1} \\ \Leftrightarrow \tau_1 &\leq \tau'_1. \end{aligned} \quad (4.32)$$

Consequently, the clock-like tree, which allows only independent evolution of adjacent edges is not network identifiable from our convergence-divergence network, where the two edges evolve independently for a longer period of time, before some convergence between the two edges happens.

Comparing our three two-taxon trees and networks, each tree and network has only a single variable phylogenetic tensor element,  $q_{11}$ . For all trees and networks, all other elements of the phylogenetic tensors do not vary with the tree or network. For the networks to be identifiable the system must not be underdetermined. This means that since there is only one variable phylogenetic tensor element, for identifiability there must be only a single time parameter. Consequently,

the clock-like tree is identifiable, while the non-clock-like tree and the convergence-divergence network are not identifiable. It has also been shown that the minimum length interval for the variable element is  $0 < q_{11} \leq 1$  for all three networks. We can therefore conclude that none of the networks are distinguishable from each other. The convergence-divergence network is not network identifiable from the clock-like tree, with divergence followed by convergence equivalent to less divergence with no convergence. Given that the clock-like tree is the only identifiable network and it is the least parameter rich of the networks, the two-taxon clock-like tree should generally be preferred under the binary symmetric model.

For some networks with more than two taxa, we will see that the two-taxon convergence-divergence network is embedded as a subnetwork inside the network. An issue that arises is determining whether these networks are distinguishable from similar clock-like networks where the two-taxon convergence-divergence network embedded in the network is replaced with the two-taxon clock-like tree. For example, consider the three trees and networks in Figure 4.4 below.

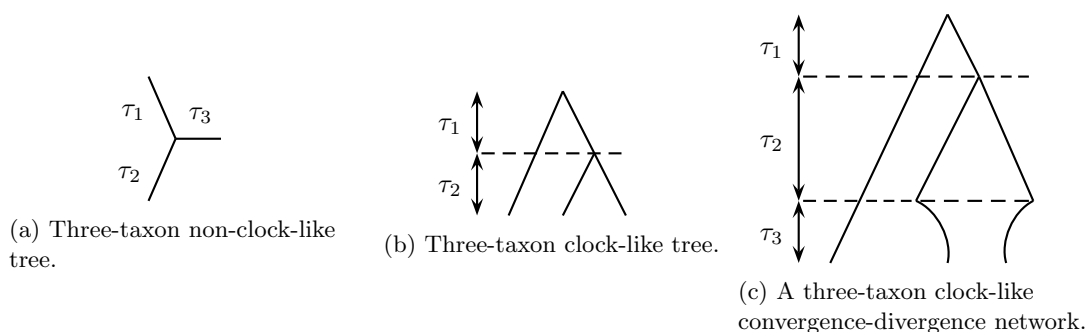


Figure 4.4: Three three-taxon trees and networks.

We would like to know whether any of these trees and networks are distinguishable from each other. From the results for the two-taxon case we might expect the three-taxon clock-like tree and the three-taxon clock-like convergence-divergence network shown above to be indistinguishable from each other. To see whether this is the case we must compare the two-taxon clock-like tree embedded in the three-taxon clock-like tree to the two-taxon clock-like convergence-divergence network embedded in the three-taxon clock-like convergence-divergence network. The tree and network that we will compare are shown in Figure 4.5 below.

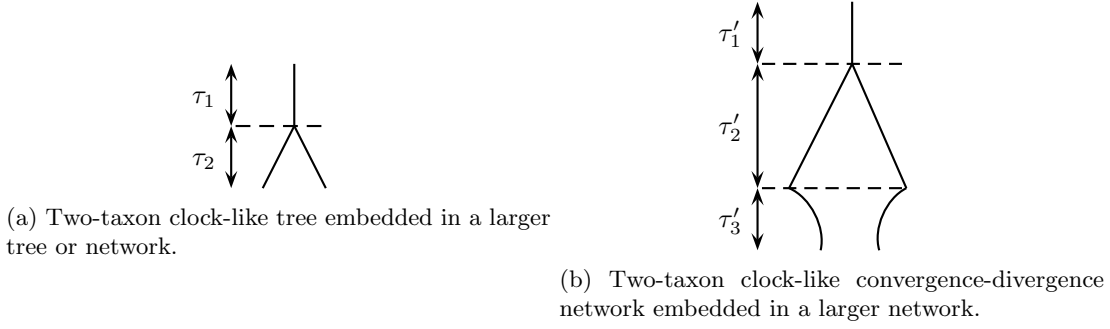


Figure 4.5: Two-taxon clock-like tree and two-taxon clock-like convergence-divergence network embedded in a larger network.

Recall from the two-taxon case that the tree and network are not network identifiable provided  $\tau_2 \leq \tau'_2$ . Consequently,

$$\tau_2 \leq \tau'_2 + \tau'_3. \quad (4.33)$$

If the tree and network are to be embedded in a larger tree or network, then we require the sum of the edge lengths to be equal,

$$\tau_1 + \tau_2 = \tau'_1 + \tau'_2 + \tau'_3. \quad (4.34)$$

Rearranging in terms of  $\tau_1$ ,

$$\tau_1 = \tau'_1 + \tau'_2 + \tau'_3 - \tau_2 \geq \tau'_1. \quad (4.35)$$

This means that we can always find a clock-like tree, with the parameters,  $\{\tau_1, \tau_2\}$ , that is not network identifiable from the clock-like convergence-divergence network, with the parameters,  $\{\tau'_1, \tau'_2, \tau'_3\}$ , embedded in a larger convergence-divergence network. The clock-like tree and clock-like convergence-divergence network will not be distinguishable from each other either.

**4.2.4.0.1** We can conclude that convergence-divergence networks for which the two-taxon convergence-divergence network is embedded in will not be identifiable, nor will they be network identifiable nor distinguishable from the same convergence-divergence network with the embedded two-taxon convergence-divergence network replaced by the two-taxon clock-like tree.

### 4.3 Three-Taxon Phylogenetic Tensors

We saw that of the three two-taxon trees and networks under the binary symmetric model, only the clock-like tree was a suitable candidate tree or network. We will now compare some three-taxon trees and networks: the non-clock-like tree, the clock-like tree and the clock-like

convergence-divergence network with a convergence period involving the two non-sister taxa. It should be noted that there are many more networks which we will compare. For these other networks we will simply present the results without deriving the phylogenetic tensor constraints.

### 4.3.1 Non-Clock-Like Tree

The first three-taxon tree that we will consider is the non-clock-like tree, shown below in Figure 4.6.

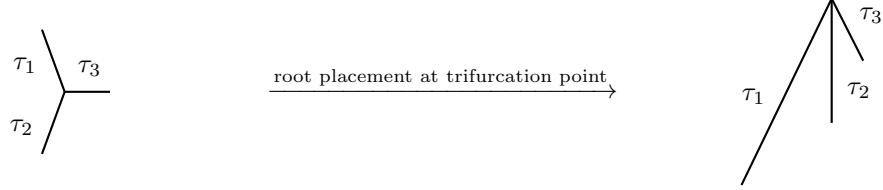


Figure 4.6: Three-taxon non-clock-like tree.

According to the procedure from Chapter 3, the phylogenetic tensor for this tree is

$$\hat{P} = e^{\frac{1}{2}(\hat{R}_{100}\tau_1 + \hat{R}_{010}\tau_2 + \hat{R}_{001}\tau_3)} \cdot \hat{\Pi} = \begin{bmatrix} q_{000} \\ q_{001} \\ q_{010} \\ q_{011} \\ q_{100} \\ q_{101} \\ q_{110} \\ q_{111} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ e^{-(\tau_2 + \tau_3)} \\ 0 \\ e^{-(\tau_1 + \tau_3)} \\ e^{-(\tau_1 + \tau_2)} \\ 0 \end{bmatrix}. \quad (4.36)$$

Demanding  $\tau_1, \tau_2, \tau_3 \geq 0$ , we can see that

$$0 < e^{-(\tau_2 + \tau_3)} \leq 1, \quad 0 < e^{-(\tau_1 + \tau_3)} \leq 1, \quad 0 < e^{-(\tau_1 + \tau_2)} \leq 1. \quad (4.37)$$

Hence  $0 < q_{011}, q_{101}, q_{110} \leq 1$  are the tight intervals treated independently containing all possible values of the phylogenetic tensor elements.

By using simple algebraic operations, we can solve for the three numerical parameters as functions of the phylogenetic tensor elements,

$$\begin{cases} \tau_1 &= \ln \frac{q_{011}}{q_{101}q_{110}}, \\ \tau_2 &= \ln \frac{q_{101}}{q_{011}q_{110}}, \\ \tau_3 &= \ln \frac{q_{110}}{q_{011}q_{101}}. \end{cases} \quad (4.38)$$

The natural logarithm is a bijective function which maps the phylogenetic tensor elements to the numerical parameters, provided that the phylogenetic tensor elements are adequately constrained. We can conclude that the non-clock-like tree is identifiable.

From the solutions for the numerical parameters, demanding that each  $\tau_i \geq 0$ , the set of constraints on the phylogenetic tensor is

$$\{q_{011} \geq q_{101}q_{110}, \quad q_{101} \geq q_{011}q_{110}, \quad q_{110} \geq q_{011}q_{101}\}. \quad (4.39)$$

Since there is no molecular clock or sense of direction of time for the non-clock-like tree, permuting the taxa does not impact on the constraints.

### 4.3.2 Clock-Like Tree

The next tree to consider is the three-taxon clock-like tree, shown in Figure 4.7 below.

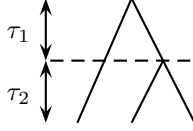


Figure 4.7: Three-taxon clock-like tree.

The phylogenetic tensor for this tree is

$$\hat{P} = e^{\frac{1}{2}(\hat{R}_{100} + \hat{R}_{010} + \hat{R}_{001})\tau_2} \cdot e^{\frac{1}{2}(\hat{R}_{100} + \hat{R}_{011})\tau_1} \cdot \hat{\Pi} = \begin{bmatrix} q_{000} \\ q_{001} \\ q_{010} \\ q_{011} \\ q_{100} \\ q_{101} \\ q_{110} \\ q_{111} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ e^{-2\tau_2} \\ 0 \\ e^{-2(\tau_1 + \tau_2)} \\ e^{-2(\tau_1 + \tau_2)} \\ 0 \end{bmatrix}. \quad (4.40)$$

Similarly to the non-clock-like tree, the clock-like tree is also identifiable.

Demanding  $\tau_1, \tau_2 \geq 0$ ,

$$0 < e^{-2\tau_2} \leq 1, \quad 0 < e^{-2(\tau_1 + \tau_2)} \leq 1. \quad (4.41)$$

Hence  $0 < q_{011}, q_{101}, q_{110} \leq 1$  are the tight intervals treated independently containing all possible values of the phylogenetic tensor elements.

We can find the constraints on the phylogenetic tensor by solving for the numerical parameters as functions of the phylogenetic tensor elements. However, we can also find the constraints simply by inspecting the phylogenetic tensor. By inspection, it can be seen that the full set of constraints

on the phylogenetic tensor is

$$\{q_{101} = q_{110}, \quad q_{011} \geq q_{110}\}. \quad (4.42)$$

### 4.3.3 Clock-Like Convergence-Divergence Network

We will look at only one example of a three-taxon convergence-divergence network here. The example we will look at is the three-taxon clock-like tree with convergence between the two non-sister taxa, shown in Figure 4.8 below. We will later discuss other examples of networks that fit our criteria for convergence-divergence networks detailed in Chapter 3.

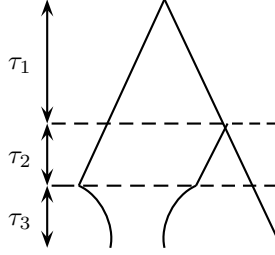


Figure 4.8: Three-taxon clock-like convergence-divergence network with convergence between non-sister taxa.

The phylogenetic tensor for this network is

$$\begin{aligned} \hat{P} &= e^{\frac{1}{2}(\hat{R}_{110} + \hat{R}_{001})\tau_3} \cdot e^{\frac{1}{2}(\hat{R}_{100} + \hat{R}_{010} + \hat{R}_{001})\tau_2} \cdot e^{\frac{1}{2}(\hat{R}_{100} + \hat{R}_{011})\tau_1} \cdot \hat{\Pi} \\ &= \begin{bmatrix} q_{000} \\ q_{001} \\ q_{010} \\ q_{011} \\ q_{100} \\ q_{101} \\ q_{110} \\ q_{111} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ e^{-2(\tau_2 + \tau_3)} \\ 0 \\ e^{-2(\tau_1 + \tau_2 + \tau_3)} \\ 1 - e^{-\tau_3} (1 - e^{-2(\tau_1 + \tau_2)}) \\ 0 \end{bmatrix}. \end{aligned} \quad (4.43)$$

Using a similar argument to that used for the two-taxon convergence-divergence network and demanding  $\tau_1, \tau_2, \tau_3 \geq 0$ ,

$$0 < e^{-2(\tau_2 + \tau_3)} \leq 1, \quad 0 < e^{-2(\tau_1 + \tau_2 + \tau_3)} \leq 1, \quad 0 < 1 - e^{-\tau_3} (1 - e^{-2(\tau_1 + \tau_2)}) \leq 1. \quad (4.44)$$

Hence  $0 < q_{011}, q_{101}, q_{110} \leq 1$  are the tight intervals treated independently containing all possible values of the phylogenetic tensor elements.

To find the full set of constraints on the phylogenetic tensor we will express the time parameters as functions of the phylogenetic tensor elements. Since the constraints on the time



parameters are known,  $\tau_1, \tau_2, \tau_3 \geq 0$ , we can then find the constraints on the phylogenetic tensor elements involving multiple elements. We will first make the substitutions,  $e^{-\tau_i} = x_i$ , for  $i = 1, 2, 3$ . Solving for each  $x_i$  and choosing solutions that allow for non-negative parameters when there is a choice, we find

$$\begin{cases} x_1 &= \sqrt{\frac{q_{101}}{q_{011}}}, \\ x_2 &= \frac{\sqrt{q_{011}} \left( -(1 - q_{110}) + \sqrt{(1 - q_{110})^2 + 4q_{101}} \right)}{2q_{101}}, \\ x_3 &= \frac{1}{2} \left( 1 - q_{110} + \sqrt{(1 - q_{110})^2 + 4q_{101}} \right). \end{cases} \quad (4.45)$$

We find that the convergence-divergence network with convergence between non-sister taxa is identifiable. We now demand that  $0 < x_1, x_2, x_3 \leq 1$ , since  $x_1, x_2$  and  $x_3$  are all functions of the time parameters. We then determine what constraints  $q_{011}$ ,  $q_{101}$  and  $q_{110}$  must have.

We start by demanding  $x_1 > 0$ . Since  $q_{011}, q_{101} \geq 0$ , this does not introduce any new constraints on the phylogenetic tensor elements.

Demanding  $x_1 \leq 1$ ,

$$q_{011} \geq q_{101}, \quad (4.46)$$

since  $q_{011}, q_{101} \geq 0$ .

Since  $q_{101} > 0$ , demanding  $x_2 > 0$ ,

$$\sqrt{q_{011}} \left( -(1 - q_{110}) + \sqrt{(1 - q_{110})^2 + 4q_{101}} \right) > 0. \quad (4.47)$$

Since  $q_{011} > 0$ ,

$$\begin{aligned} -(1 - q_{110}) + \sqrt{(1 - q_{110})^2 + 4q_{101}} &> 0 \\ \Leftrightarrow \sqrt{(1 - q_{110})^2 + 4q_{101}} &> (1 - q_{110}). \end{aligned} \quad (4.48)$$

Since both sides of the inequality must be non-negative and the argument inside the square root must be positive,

$$\begin{aligned} (1 - q_{110})^2 + 4q_{101} &> (1 - q_{110})^2 \\ \Leftrightarrow 4q_{101} &> 0 \\ \Leftrightarrow q_{101} &> 0. \end{aligned} \quad (4.49)$$

We already knew that  $q_{101} > 0$ . Hence, demanding  $x_2 > 0$  does not introduce any new constraints on the phylogenetic tensor elements.

Demanding  $x_2 \leq 1$ ,

$$\begin{aligned}
\sqrt{q_{011}} \left( -(1 - q_{110}) + \sqrt{(1 - q_{110})^2 + 4q_{101}} \right) &\leq 2q_{101} \\
\Leftrightarrow -(1 - q_{110}) + \sqrt{(1 - q_{110})^2 + 4q_{101}} &\leq \frac{2q_{101}}{\sqrt{q_{011}}} \\
\Leftrightarrow \sqrt{(1 - q_{110})^2 + 4q_{101}} &\leq (1 - q_{110}) + \frac{2q_{101}}{\sqrt{q_{011}}}.
\end{aligned} \tag{4.50}$$

Since both sides of the inequality must be non-negative and the argument inside the square root must be positive, the inequality remains after we square both sides,

$$\begin{aligned}
(1 - q_{110})^2 + 4q_{101} &\leq (1 - q_{110})^2 + \frac{4q_{101}^2}{q_{011}} + \frac{4q_{101}(1 - q_{110})}{\sqrt{q_{011}}} \\
\Leftrightarrow 4q_{101} &\leq \frac{4q_{101}^2}{q_{011}} + \frac{4q_{101}(1 - q_{110})}{\sqrt{q_{011}}} \\
\Leftrightarrow 1 &\leq \frac{q_{101}}{q_{011}} + \frac{(1 - q_{110})}{\sqrt{q_{011}}} \\
\Leftrightarrow q_{011} &\leq q_{101} + (1 - q_{110})\sqrt{q_{011}} \\
\Leftrightarrow q_{011} - q_{101} &\leq (1 - q_{110})\sqrt{q_{011}}.
\end{aligned} \tag{4.51}$$

Since  $q_{011} \geq q_{101}$ , we can square both sides and the equality will remain,

$$(q_{011} - q_{101})^2 \leq (1 - q_{110})^2 q_{011}. \tag{4.52}$$

We now demand  $x_3 > 0$ . Recall that we have already established that  $q_{101} > 0$  and  $q_{110} \leq 1$ . Hence  $1 - q_{110} \geq 0$  and

$$\frac{1}{2} \left( 1 - q_{110} + \sqrt{(1 - q_{110})^2 + 4q_{101}} \right) > 0. \tag{4.53}$$

Hence, imposing this constraint does not introduce any new constraints to the phylogenetic tensor elements.

Finally, demanding  $x_3 \leq 1$ ,

$$\begin{aligned}
\frac{1}{2} \left( 1 - q_{110} + \sqrt{(1 - q_{110})^2 + 4q_{101}} \right) &\leq 1 \\
\Leftrightarrow 1 - q_{110} + \sqrt{(1 - q_{110})^2 + 4q_{101}} &\leq 2 \\
\Leftrightarrow \sqrt{(1 - q_{110})^2 + 4q_{101}} &\leq 1 + q_{110}.
\end{aligned} \tag{4.54}$$

Since both sides of the inequality must be positive and real, the inequality remains after we

square both sides,

$$\begin{aligned}
 (1 - q_{110})^2 + 4q_{101} &\leq 1 + 2q_{110} + q_{110}^2 \\
 \Leftrightarrow 1 - 2q_{110} + q_{110}^2 + 4q_{101} &\leq 1 + 2q_{110} + q_{110}^2 \\
 \Leftrightarrow q_{101} &\leq q_{110}.
 \end{aligned} \tag{4.55}$$

So the constraint  $q_{101} \leq q_{110}$  arises from demanding  $x_3 = e^{-\tau_3} \leq 1$  or  $\tau_3 > 0$ .

Hence, the full set of constraints on the phylogenetic tensor is

$$\left\{ q_{011} \geq q_{101}, \quad q_{110} \geq q_{101}, \quad q_{011} (1 - q_{110})^2 \geq (q_{011} - q_{101})^2 \right\}. \tag{4.56}$$

We can conclude that if a phylogenetic tensor satisfies this set of constraints, we can find the time parameters,  $\tau_1$ ,  $\tau_2$  and  $\tau_3$ , on a convergence-divergence network with convergence between non-sister taxa. Likewise, if we start with a phylogenetic tensor and time parameters from the convergence-divergence network with convergence between non-sister taxa, this set of constraints must be satisfied.

We saw that all three of the three-taxon trees and networks that we have so far examined are identifiable.

The constant elements of the phylogenetic tensors must be the same for all three of the three-taxon trees and networks. These constraints are

$$\{q_{000} = 1, \quad q_{001} = q_{010} = q_{100} = q_{111} = 0\}. \tag{4.57}$$

The non-constant elements of the phylogenetic tensors must all satisfy the constraints,

$$\{0 < q_{011}, q_{101}, q_{110} \leq 1\}. \tag{4.58}$$

We can conclude that these three trees and networks are all distinguishable from each other since the sets of constraints are not identical. Recall from our definition of distinguishability that two networks are distinguishable if their probability spaces are not identical. This does not preclude a particular set of phylogenetic tensor elements satisfying the constraints for multiple distinguishable trees and networks. What it does mean is that there must be at least one particular pattern frequency that only satisfies the constraints of one of the trees or networks. The clock-like tree, however, has one less numerical parameter than the non-clock-like tree and the clock-like convergence-divergence network with convergence between non-sister taxa. Hence, from a practical model-fitting perspective it should be given preference unless there is a strong argument against the molecular clock assumption or tree behaviour or if the non-clock-like tree or the clock-like convergence-divergence network with convergence between non-sister taxa provide a significantly better fit than the clock-like tree.

#### 4.3.4 Comparisons of Other Three-Taxon Networks

Having defined our restrictions on the convergence periods, we will now analyse the remaining three-taxon networks. There are nine trees and networks which meet our restrictions on the convergence periods. We will label these as shown below in Figure 4.9.

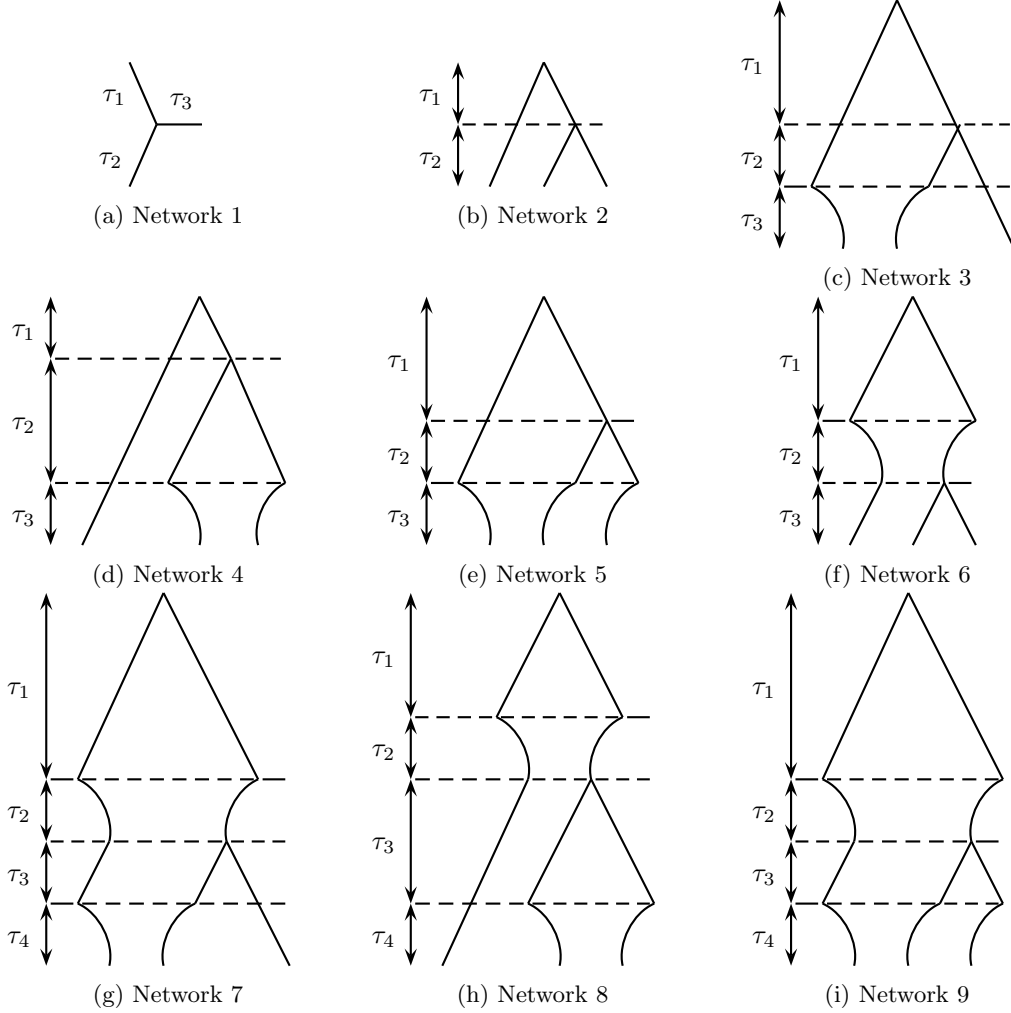


Figure 4.9: Some three-taxon networks. Networks 1, 2 and 3 have been introduced already. Network 1 is the non-clock-like tree. All other trees and networks are clock-like.

Recall from Section 4.1 on page 69 that the first transformed phylogenetic tensor element is the conservation of probability,  $q_{00\dots 0} = 1$ . Recall also that any transformed phylogenetic tensor element with the sum of its indices being odd must be identically zero. Hence, for any three-taxon tree or network,  $q_{001} = q_{010} = q_{100} = q_{111} = 0$ . Recall also that any transformed phylogenetic tensor element with the sum of its indices being even will be non-zero. Hence,  $q_{011}, q_{101}, q_{110} \neq 0$ . There are only  $2^{n-1} - 1 = 3$  variable phylogenetic tensor elements for any three-taxon tree or

network. As a result, any trees or networks with more than three time parameters will not be identifiable. We can immediately rule out Network 7, Network 8 and Network 9 as they cannot possibly be identifiable. From our two-taxon result, Network 2, Network 4, Network 6 and Network 8 are not distinguishable from each other, Network 3 and Network 7 are not distinguishable from each other and Network 5 and Network 9 are not distinguishable from each other. Hence, we will focus only on Network 1, Network 2, Network 3 and Network 5 as they are the least parameter rich. The four trees and networks remaining to examine are the non-clock-like tree, the clock-like tree, the clock-like convergence-divergence network with convergence between non-sister taxa and the clock-like convergence-divergence network with convergence between all three taxa.

To determine whether a tree or network is distinguishable from each other we will examine the intersection of their probability spaces, that is the intersection of the sets of constraints on the transformed phylogenetic tensor elements. If the two sets of constraints are identical then the two trees or networks will not be distinguishable from each other. If the two sets of constraints are not identical then the two trees or networks will be distinguishable from each other.

#### 4.3.4.1 Network 5

Before analysing the identifiability of these trees and networks and whether they are distinguishable from one another, we need to find the constraints for the clock-like convergence-divergence network with convergence between all three taxa, shown in Figure 4.10 below.

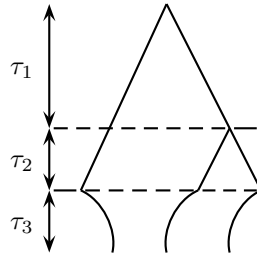


Figure 4.10: Three-taxon clock-like convergence-divergence network with convergence between all three taxa.

The phylogenetic tensor for this network is

$$\begin{aligned} \hat{P} = e^{\frac{1}{2}\hat{R}_{111}\tau_3} \cdot e^{\frac{1}{2}(\hat{R}_{100}+\hat{R}_{010}+\hat{R}_{001})\tau_2} \cdot e^{\frac{1}{2}(\hat{R}_{100}+\hat{R}_{011})\tau_1} \cdot \hat{\Pi} = & \begin{bmatrix} q_{000} \\ q_{001} \\ q_{010} \\ q_{011} \\ q_{100} \\ q_{101} \\ q_{110} \\ q_{111} \end{bmatrix} \\ = & \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 - e^{-\tau_3} (1 - e^{-2\tau_2}) \\ 0 \\ 1 - e^{-\tau_3} (1 - e^{-2(\tau_1+\tau_2)}) \\ 1 - e^{-\tau_3} (1 - e^{-2(\tau_1+\tau_2)}) \\ 0 \end{bmatrix}. \end{aligned} \quad (4.59)$$

Using a similar argument to that used previously and demanding  $\tau_1, \tau_2, \tau_3 \geq 0$ ,

$$0 < 1 - e^{-\tau_3} (1 - e^{-2\tau_2}), \quad 0 < 1 - e^{-\tau_3} (1 - e^{-2(\tau_1+\tau_2)}) \leq 1. \quad (4.60)$$

Hence  $0 < q_{011}, q_{101}, q_{110} \leq 1$  are the minimum length intervals containing all possible values of the individual tensor elements.

By inspection, it can be seen that the full set of constraints on the phylogenetic tensor is

$$\{q_{101} = q_{110}, \quad q_{011} \geq q_{110}\}. \quad (4.61)$$

Clearly this network is not identifiable and cannot be distinguished from the three-taxon clock-like tree. We conclude that the three-taxon clock-like tree is preferable to the three-taxon clock-like convergence-divergence network with convergence between all three taxa.

#### 4.3.5 Distinguishability of Three-Taxon Trees and Networks

We have established that there are three remaining three-taxon trees and networks that are identifiable and distinguishable from each other: the non-clock-like tree, the clock-like tree and the clock-like convergence-divergence network with convergence between non-sister taxa. We have constructed Table 4.1 below, which displays whether or not a particular constraint is met for those three trees and networks, as well as the other six networks that we originally considered.

In addition to the constraints displayed in the table, there are three other constraints which must be met for the non-clock-like tree.

Network(s)	$q_{101} = q_{110}$ (Y/N)	$q_{011} \geq q_{101}$ (Y/N)	$q_{011}(1 - q_{110})^2 \geq (q_{011} - q_{101})^2$ (Y/N)
1	N	N	N
2, 4, 5, 6, 8, 9	Y	Y	N
3, 7	N	Y	Y

Table 4.1: Summary of network constraints which must be met. “Y” indicates that the constraint is necessary to be met for that particular network, while “N” indicates that the constraint is not necessary to be met for the network in question.

In addition, the non-clock-like tree, or Network 1, must meet the constraints,

$$\{q_{011} \geq q_{101}q_{110}, \quad q_{101} \geq q_{011}q_{110}, \quad q_{110} \geq q_{011}q_{101}\}. \quad (4.62)$$

It should be noted that this table expresses minimum constraints that must be met for the given tree or network. If meeting a given constraint is not necessary for a particular network that constraint could still be met for the network. For example, the constraint  $q_{101} = q_{110}$  being met does not discount the non-clock-like tree as a candidate, however the constraint  $q_{101} = q_{110}$  not being met does discount the clock-like tree as a candidate.

The probability spaces that the networks occupy will be denoted by  $\Omega_1$ ,  $\Omega_2$  and  $\Omega_3$ , with the subscripts referring to the three networks in question as labelled earlier. We will look at the intersections of every pair of probability spaces.

#### 4.3.5.1 Network 1 and Network 2

We will start with the intersection of the probability spaces of the two trees,  $\Omega_1 \cap \Omega_2$ . Recall that the constraints for Network 2 are

$$\{q_{101} = q_{110}, \quad q_{011} \geq q_{110}\}. \quad (4.63)$$

$q_{110}$  is not a free element and can be replaced by  $q_{101}$  for every constraint for Network 1. Recall that the constraints for Network 1 are

$$\{q_{011} \geq q_{101}q_{110}, \quad q_{101} \geq q_{011}q_{110}, \quad q_{110} \geq q_{011}q_{101}\}. \quad (4.64)$$

Replacing  $q_{110}$  with  $q_{101}$ , the constraints become

$$\{q_{011} \geq q_{101}^2, \quad 1 \geq q_{011}, \quad 1 \geq q_{011}\}. \quad (4.65)$$

From the second constraint for Network 2 and since  $1 \geq q_{101} \geq q_{101}^2$ , we can conclude that  $q_{011} \geq q_{101}^2$ . The intersection of the two probability spaces must then be the probability space

for Network 2,

$$\begin{aligned}\Omega_1 \cap \Omega_2 &= \Omega_2, \\ \Omega_2 &\subset \Omega_1.\end{aligned}\tag{4.66}$$

$\Omega_2$  is a proper subset of  $\Omega_1$  since it does not have the same set of constraints as this tree and therefore does not have the same probability space.

#### 4.3.5.2 Network 2 and Network 3

We will now examine the intersection of the probability spaces of Network 2 and Network 3,  $\Omega_2 \cap \Omega_3$ . Recall the constraints for Network 3 are

$$\left\{ q_{011} \geq q_{101}, \quad q_{110} \geq q_{101}, \quad q_{011}(1 - q_{110})^2 \geq (q_{011} - q_{101})^2 \right\}.\tag{4.67}$$

Clearly the two networks have the constraint,  $q_{011} \geq q_{101}$ , in common. Recall that  $q_{110}$  is not a free element for Network 2 and can be replaced by  $q_{101}$  for every constraint for Network 3. Replacing  $q_{110}$  with  $q_{101}$ , the constraints become

$$\left\{ q_{011} \geq q_{101}, \quad q_{101} \geq q_{101}, \quad q_{011}(1 - q_{101})^2 \geq (q_{011} - q_{101})^2 \right\}.\tag{4.68}$$

Clearly the second constraint is trivial. The only constraint left to examine is the last constraint, which when expanded out becomes

$$\begin{aligned}q_{011} + q_{011}q_{101}^2 - 2q_{011}q_{101} &\geq q_{011}^2 + q_{101}^2 - 2q_{011}q_{101} \\ \Leftrightarrow q_{011} + q_{011}q_{101}^2 &\geq q_{011}^2 + q_{101}^2 \\ \Leftrightarrow q_{011}(1 - q_{011}) &\geq q_{101}^2(1 - q_{011}).\end{aligned}\tag{4.69}$$

Since  $q_{011} \neq 1$  generally,

$$q_{011} \geq q_{101}^2.\tag{4.70}$$

From the second constraint for Network 2 and since  $1 \geq q_{101} \geq q_{101}^2$ , we can conclude that  $q_{011} \geq q_{101}^2$ . The intersection of the two probability spaces must then be the probability space for Network 2,

$$\begin{aligned}\Omega_2 \cap \Omega_3 &= \Omega_2, \\ \Omega_2 &\subset \Omega_3.\end{aligned}\tag{4.71}$$

$\Omega_2$  is a proper subset of  $\Omega_3$  since it does not have the same set of constraints as this tree and therefore does not have the same probability space.

We have shown that if the constraints for the clock-like tree are met, the constraints for the



non-clock-like tree and the constraints for the clock-like convergence-divergence network must also be met. This is to be expected since the clock-like tree can be recovered from both the non-clock-like tree and the clock-like convergence-divergence network with the appropriate time parameters. Hence,

$$\begin{aligned}\Omega_2 &\subset \Omega_1, \\ \Omega_2 &\subset \Omega_3.\end{aligned}\tag{4.72}$$

$\Omega_2$  is a proper subset of both  $\Omega_1$  and  $\Omega_3$  since it does not have the same set of constraints as these two trees and networks and therefore does not have the same probability space.

#### 4.3.5.3 Network 1 and Network 3

The last intersection of probability spaces is the intersection of the probability spaces of Network 1 and Network 3,  $\Omega_1 \cap \Omega_3$ . The set of constraints in the region will be

$$\begin{aligned}\Omega_1 \cap \Omega_3 = \{ &q_{011} \geq q_{101}q_{110}, \quad q_{101} \geq q_{011}q_{110}, \quad q_{110} \geq q_{011}q_{101}, \quad q_{011} \geq q_{101}, \quad q_{110} \geq q_{101}, \\ &q_{011}(1 - q_{110})^2 \geq (q_{011} - q_{101})^2\}.\end{aligned}\tag{4.73}$$

Some of these constraints are implied by others. We will now find out which of these constraints are implied by the other constraints. Since  $0 < q_{110} \leq 1$ ,

$$q_{011} \geq q_{101} \Rightarrow q_{011} \geq q_{101}q_{110}.\tag{4.74}$$

Since  $0 < q_{011} \leq 1$ ,

$$q_{110} \geq q_{101} \Rightarrow q_{110} \geq q_{011}q_{101}.\tag{4.75}$$

Hence, the constraints  $q_{011} \geq q_{101}q_{110}$  and  $q_{110} \geq q_{011}q_{101}$  are implied by the other constraints. Since  $0 < q_{011}, q_{101}, q_{110} \leq 1$ ,

$$\begin{aligned}q_{101} \geq q_{011}q_{110} &\Leftrightarrow \frac{q_{101}}{q_{011}} \geq q_{110} \\ &\Leftrightarrow \frac{q_{101}^2}{q_{011}^2} \geq q_{110}^2 \\ &\Leftrightarrow -q_{110}^2 \geq -\frac{q_{101}^2}{q_{011}^2} \\ &\Leftrightarrow 1 - q_{110}^2 \geq 1 - \frac{q_{101}^2}{q_{011}^2}.\end{aligned}\tag{4.76}$$

Since  $q_{101} \geq q_{011}q_{110}$ ,

$$\begin{aligned} \frac{q_{101}}{q_{011}} &\geq q_{110} \\ \Leftrightarrow -2q_{110} &\geq -2\frac{q_{101}}{q_{011}}. \end{aligned} \tag{4.77}$$

Combining the two inequalities,  $1 - q_{110}^2 \geq 1 - \frac{q_{101}^2}{q_{011}^2}$  and  $-2q_{110} \geq -2\frac{q_{101}}{q_{011}}$ ,

$$\begin{aligned} 1 - 2q_{110} - q_{110}^2 &\geq 1 - 2\frac{q_{101}}{q_{011}} - \frac{q_{101}^2}{q_{011}^2} \\ \Leftrightarrow (1 - q_{110})^2 &\geq \left(1 - \frac{q_{101}}{q_{011}}\right)^2 \\ \Rightarrow (1 - q_{110})^2 &\geq q_{011} \left(1 - \frac{q_{101}}{q_{011}}\right)^2 \\ \Leftrightarrow (1 - q_{110})^2 &\geq q_{011} \left(\frac{q_{011} - q_{101}}{q_{011}}\right)^2 \\ \Leftrightarrow q_{011} (1 - q_{110})^2 &\geq (q_{011} - q_{101})^2. \end{aligned} \tag{4.78}$$

This is one of the constraints in the probability space for Network 3. Finally, the set can be simplified to

$$\Omega_1 \cap \Omega_3 = \{q_{011} \geq q_{101}, \quad q_{110} \geq q_{101}, \quad q_{101} \geq q_{011}q_{110}\}. \tag{4.79}$$

Consequently, there must be three distinct regions involving the non-clock-like tree and the clock-like convergence-divergence network. There must be a region where the constraints of both networks are met and regions where the constraints of only one network are met. Below in Figure 4.11 is a diagram representing the four regions in the probability space involving the three trees and networks. The diagram is not to scale, nor do the shapes of the regions have any meaning other than the reflection that some regions border each other in the probability space.

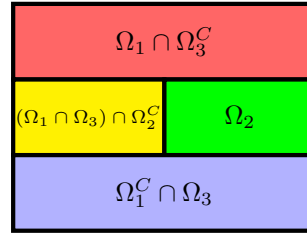


Figure 4.11: Probability spaces of the networks.  $C$  in the superscripts refers to the complement of the set. In other words, the constraints for that network are not met in that region. In the region outside of the probability space formed by the union of the probability spaces for the three trees and networks, none of the sets of constraints of the trees or networks examined are met. In part of this region the constraints for more complex models than the binary symmetric model will be met. For example, for the binary symmetric model,  $p_{\mathbf{i}} = p_{\bar{\mathbf{i}}}$ , where  $\mathbf{i}$  is an element of the phylogenetic tensor and  $\bar{\mathbf{i}}$  is the term in the phylogenetic tensor where every ‘0’ has been replaced by a ‘1’ and every ‘1’ has been replaced by a ‘0’. If these constraints are not met then the constraints on a more complex model could still be met.

We can write a pseudocode algorithm to determine which tree or network a set of constraints could have arisen on. Below is the pseudocode algorithm.

```

if Network 1 constraints == TRUE then
  if Network 3 constraints == TRUE then
    if Network 2 constraints == TRUE then
      tree or network  $\leftarrow$  Network 1, Network 2 or Network 3
    else
      tree or network  $\leftarrow$  Network 1 or Network 3
    end if
  else
    tree or network  $\leftarrow$  Network 1
  end if
else
  tree or network  $\leftarrow$  Network 3
end if

```

In summary, there are four interesting regions in the probability space of the networks. A phylogenetic tensor either belongs to the non-clock-like tree exclusively, the clock-like convergence-divergence network exclusively, either of the non-clock-like tree or the clock-like convergence-divergence network, or all three trees and networks.



## Chapter 5

# Network Identifiability of Taxon Label Permutations

In this chapter we will introduce the concept of permuting taxon labels and examine whether different taxon label permutations are network identifiable from each other on a given tree or network. Recall from Chapter 1 that a pair of trees or networks are network identifiable from each other if there is a pattern frequency that could have only arisen on one of the trees or networks. If we have non-negative time parameters on one of the trees or networks, can we find a set of non-negative time parameters on the other tree or network that gives rise to an identical phylogenetic tensor? If we can find a set of non-negative time parameters then the pair of trees or networks are not network identifiable from each other. If some of the time parameters can be negative then the pair of trees or networks are network identifiable from each other.

For the two-taxon clock-like tree and the two-taxon convergence-divergence network the choice of taxon labelling will be of no consequence since all two-taxon trees and networks are symmetrical below every node. We will start by examining the three-taxon clock-like tree, before examining the three-taxon convergence-divergence network with convergence between non-sister taxa.

### 5.1 Permuting the Taxon Labels

We now wish to see how taxon label permutations may affect the transformed phylogenetic tensor elements on a three-taxon tree or network. There are  $3! = 6$  taxon label permutations on three-taxon networks, including the original tree or network with no taxon label permutation. These permutations are described by the symmetric group,

$$S_3 = \{(), (12), (13), (23), (123), (132)\}. \quad (5.1)$$

Suppose the element,  $()$ , corresponds to the three-taxon clock-like tree shown in Figure 5.1

below.

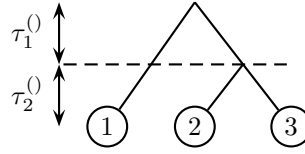


Figure 5.1: The  $()$  taxon label permutation, where the taxon labels have not been permuted.

Suppose it also corresponds to the three-taxon clock-like network shown in Figure 5.2 below.

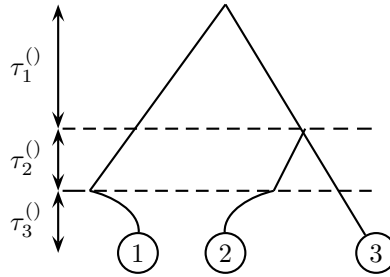


Figure 5.2: The  $()$  taxon label permutation, where the taxon labels have not been permuted.

The other elements of the symmetric group will permute this taxon labelling accordingly. Referring to Figure 5.3 on page 99, it should be noted that the  $(23)$  taxon label permutation on the three-taxon clock-like tree will not change the structure of the tree and is not particularly interesting to us. Similarly, since  $(23)(12) = (132)$ , the  $(12)$  and  $(132)$  taxon label permutations will result in the same structure on the three-taxon clock-like tree. Since  $(23)(13) = (123)$ , the  $(13)$  and  $(123)$  taxon label permutations will also result in the same structure on the three-taxon clock-like tree.

There are two issues that arise when we permute the taxon labels.

1. Suppose we permute the taxon labels. The structure of the tree or network remains unchanged. The only change that has been made is the labelling of the leaves. The tree or network must therefore have the same set of time parameters. However, the elements of the phylogenetic tensor must be permuted according to the taxon label permutation in question. For example, suppose we wish to permute the taxon labels on the first two leaves of the tree or network, counting from the left. The transformed phylogenetic tensor elements that must be equal functions are

$$q_{i_1 i_2 i_3}^{(12)}(u, v, w) = q_{i_2 i_1 i_3}^{()}(u, v, w), \quad (5.2)$$

where  $()$  refers to the original taxon labelling,  $(12)$  refers to the permuting of the first and second taxon labels, and  $u, v$  and  $w$  are any arbitrary time parameters for the convergence-divergence network.

2. Suppose we start with the site pattern frequency,  $i_1 i_2 i_3$ . We now wish to equate the transformed phylogenetic tensor elements with the same site patterns for different taxon label permutations. If we equate each element of the transformed phylogenetic tensor for the tree or network with no taxon label permutation to the tree or network with the first and second taxon labels permuted, we must have

$$q_{i_1 i_2 i_3}^{()} \left( \tau_1^{()}, \tau_2^{()}, \tau_3^{()} \right) = q_{i_1 i_2 i_3}^{(12)} \left( \tau_1^{(12)}, \tau_2^{(12)}, \tau_3^{(12)} \right). \quad (5.3)$$

The two transformed phylogenetic tensor elements will not have identical functions. We don't know a priori whether there will be any set of non-negative time parameters on one taxon label permutation that will correspond to a set of non-negative time parameters on the other taxon label permutation. The transformed phylogenetic tensor elements will only be equal for certain sets of time parameters, if any. We want to determine whether the probability distributions overlap and to find the time parameters if they do. This second issue is a question of network identifiability.

## 5.2 Determining How Often Time Parameters are Non-Negative

Suppose we have found the time parameters for a second taxon label permutation in terms of the time parameters for a first taxon label permutation on a given tree or network. Suppose that one of the time parameters on the second taxon label permutation, when expressed in terms of the non-negative time parameters on the first taxon label permutation, can take any real number. We can simulate data to determine how often the time parameter on the second taxon label permutation will be non-negative. We can generate  $x_i = e^{-\tau_i} \in U(0, 1)$  for the time parameters on the first taxon label permutation using the “runif” function in *R*. We can then evaluate the time parameter on the second taxon label permutation and determine whether it is non-negative or not. By performing many simulations we can get an idea of how often the time parameter on the second taxon label permutation will be non-negative. Below is a sample code for determining how often the time parameter on the second taxon label permutation will be non-negative.

```
# Set the number of iterations , n'.
n <- n'

# Vector for the xi=exp(-taui) in U(0,1), the i time parameters on
  the tree or network for the first taxon label permutation.
x <- vector()

# Define the number of time parameters on the tree or network for the
  first taxon label permutation, m'.
```

```

m <- m'

# Counter for the number of times that the constraint is met.
c <- 0

# Perform n iterations.
for (i in 1:n) {

# Randomly generate  $x_i = \exp(-\tau_i)$  in  $U(0,1)$  for the  $i$  time parameters
  on the tree or network for the first taxon label permutation.
  for (j in 1:m) {
    x[j] <- runif(1)
  }

# Find one of the time parameters for the second taxon label
  permutation, where  $f(x[1], x[2], \dots, x[m])$  is a function of the  $m$ 
  time parameters on the tree or network for the first taxon label
  permutation.
  t2 <- f(x[1], x[2], ..., x[m])

# If the time parameter is non-negative, increase the total count of
  non-negative time parameters by one, otherwise keep the count of
  non-negative time parameters the same.
  c <- ifelse(t2 >= 0, c+1, c)
}

# Output the number of times the time parameter is non-negative.
c

```

It should be noted that these simulations assume that each  $x_i = e^{-\tau_i}$  is uniformly distributed,  $x_i = e^{-\tau_i} \in U(0, 1)$ .

### 5.3 The Three-Taxon Clock-Like Tree

We can compare the sets of constraints on the transformed phylogenetic tensor elements for all of the taxon label permutations. Below in Figure 5.3 are the six taxon label permutations on the three-taxon clock-like tree.



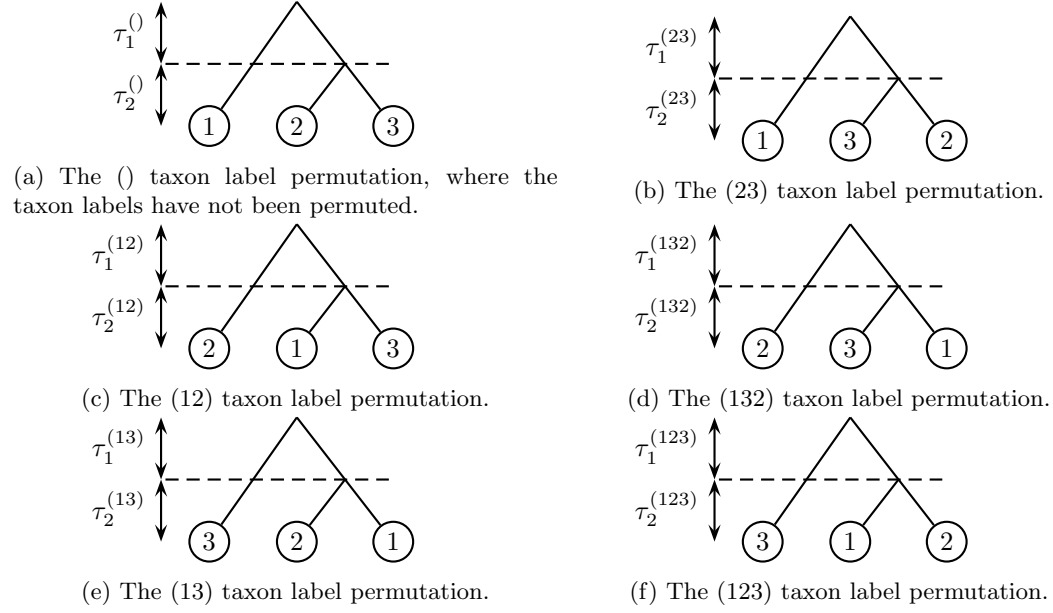


Figure 5.3: The six taxon label permutations on the three-taxon clock-like tree.

Since the order of the two taxa sharing the most recent common ancestor is inconsequential, we can see that there will be three pairs of taxon label permutations with the same set of constraints, shown in Table 5.1 below.

Taxon Label Permutations	Constraints
$()$ , $(23)$	$\left\{ q_{101}^0 = q_{110}^0, \quad q_{011}^0 \geq q_{110}^0 \right\}$
$(12)$ , $(132)$	$\left\{ q_{011}^0 = q_{110}^0, \quad q_{101}^0 \geq q_{110}^0 \right\}$
$(13)$ , $(123)$	$\left\{ q_{101}^0 = q_{011}^0, \quad q_{110}^0 \geq q_{011}^0 \right\}$

Table 5.1: Summary of constraints for each taxon label permutation.

When choosing a taxon label permutation on the clock-like tree, there are three choices for the taxon label for the non-sister taxa. Each of these three choices of taxon labellings will have two choices for the labellings of the sister taxa, which must give rise to identical phylogenetic tensors. We can see from the summary of constraints that the three choices for the taxon label for the non-sister taxa must give rise to different phylogenetic tensors. The only circumstance where two different taxon labelling choices for the non-sister taxa could have identical phylogenetic tensors is when the first time parameter is zero,  $\tau_1 = 0$ . In this circumstance the three-taxon clock-like tree becomes the tripod tree with all edges being of equal length.

Comparing taxon label permutations on the clock-like tree is straightforward, however for more complicated networks, such as the convergence-divergence network with convergence between non-sister taxa, the permuting of taxon labels is much more complicated.

## 5.4 The Convergence-Divergence Network

We will now examine the convergence-divergence network with convergence between non-sister taxa, shown in Figure 5.4 below.

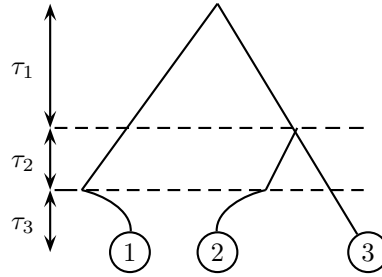


Figure 5.4: The three-taxon clock-like convergence-divergence network with convergence between non-sister taxa.

We wish to determine whether various taxon label permutations are network identifiable from each other. Recall that there are six taxon label permutations for three-taxon trees and networks. Below in Figure 5.5 are the six taxon labellings for the convergence-divergence network.

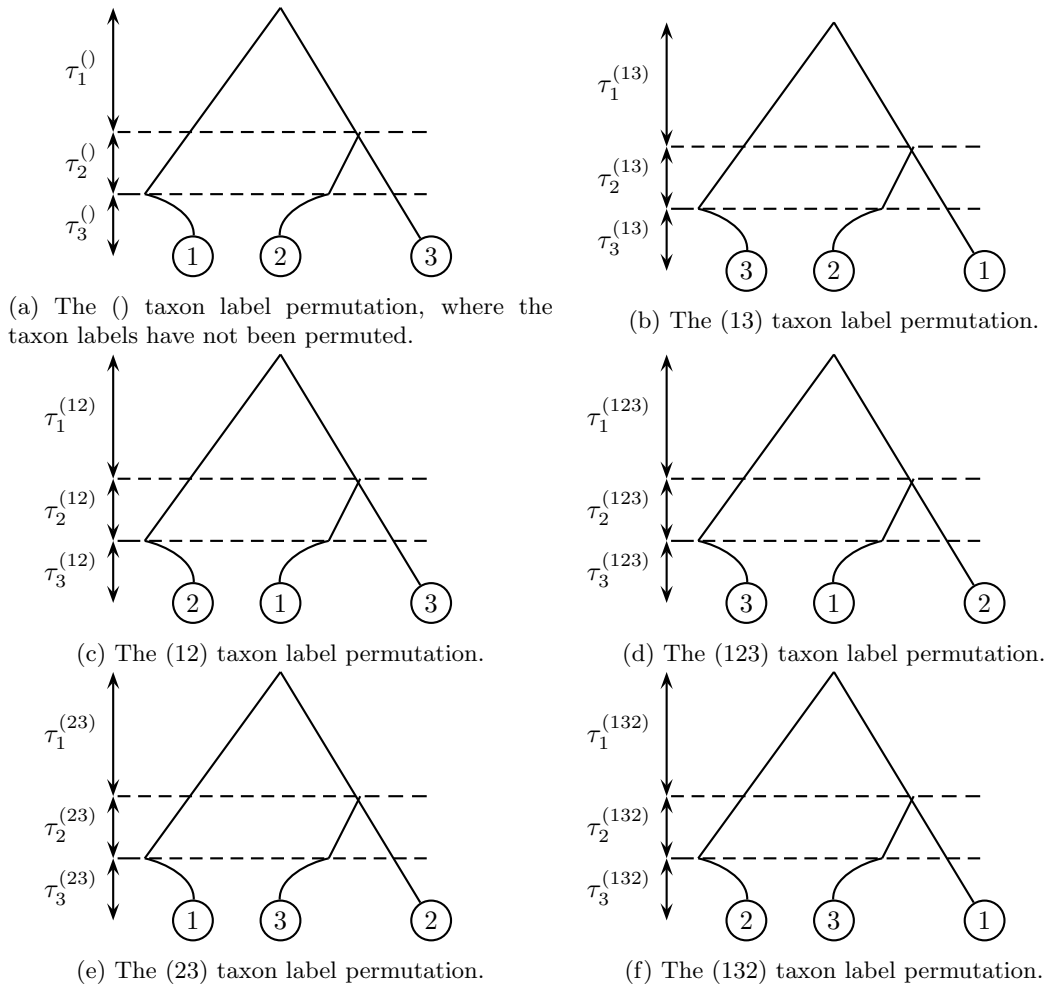


Figure 5.5: The six taxon label permutations for the three-taxon convergence-divergence network.

Unlike with the three-taxon clock-like tree, it is not immediately obvious whether the six different taxon labellings will be network identifiable from each other. To see whether the various taxon labellings are network identifiable we will solve for the time parameters for various taxon labellings in terms of the time parameters of a different taxon labelling.

## 5.5 Pairwise Distances

Before looking at a taxon label permutation, we wish to determine which pairs of taxa are “closer” to each other than other pairs on the network. We will do this by introducing the notions of *tree distance* and *pairwise distance*. Hendy and Penny [1993] introduced the notion of *tree distance* as a measure of how close two nodes are to each other on a tree. We will only use the term *tree distance* when referring to leaves on a tree.

**Definition 5.5.1 (Tree Distance).** *The **tree distance**,  $d(a, b)$ , between two leaves,  $a$  and  $b$ , on a tree is the sum of the edge lengths connecting the two leaves.*

Hendy and Penny [1993] noted that the tree distance could be found by using the Hadamard transformation. To see how this is done, let's look at the transformed phylogenetic tensor for the two-taxon clock-like tree as an example. This tree has only one variable element in the transformed phylogenetic tensor,

$$q_{11} = e^{-2\tau_1}. \quad (5.4)$$

Clearly the tree distance between the two leaves is

$$d(1, 2) = 2\tau_1 = -\ln(q_{11}). \quad (5.5)$$

We will see from the expressions for the phylogenetic tensor elements that on a tree the tree distance will simply be

$$d(a, b) = -\ln(q_A), \quad (5.6)$$

where  $A = i_1 i_2 \dots i_n$ ,  $a, b \in \{1, 2, \dots, n\}$ ,  $i_a = i_b = 1$  and  $i_k = 0$  for all  $k \in \{1, 2, \dots, n\} \setminus \{a, b\}$ .

Without considering any taxon labelling, we will now examine the distances on the three-taxon convergence-divergence network. It is immediately clear that the distance between taxon two and taxon three,  $d(2, 3)$ , is simply

$$d(2, 3) = 2(\tau_2 + \tau_3) = -\ln(q_{011}). \quad (5.7)$$

Although taxon two is converging with taxon one, it is always diverging from taxon three. It is therefore clear that the distance between taxon one and taxon three is

$$d(1, 3) = 2(\tau_1 + \tau_2 + \tau_3) = -\ln(q_{101}). \quad (5.8)$$

Unlike  $d(2, 3)$  and  $d(1, 3)$ , it is not immediately clear what  $d(1, 2)$  is intuitively. The convergence between taxon one and taxon two should decrease the distance between the two taxa. It is not immediately clear whether our definition of tree distance will be appropriate when we generalise our trees to convergence-divergence networks. We want to know whether our distance function is appropriate when convergence has occurred.

If we are to assume that the distance function is an appropriate choice, the distance between the two converging taxa on the three-taxon convergence-divergence network will be

$$d(1, 2) = -\ln(q_{110}) = -\ln\left(1 - e^{-\tau_3}\left(1 - e^{-2(\tau_1 + \tau_2)}\right)\right). \quad (5.9)$$

We will see whether this is in fact an appropriate choice for the distance between the two

converging taxa.

We will now take a moment to determine whether our distances satisfy some key criteria of a distance on a tree or network. Clearly

$$d(2, 3) = 2(\tau_2 + \tau_3) \quad (5.10)$$

and

$$d(1, 3) = 2(\tau_1 + \tau_2 + \tau_3) \quad (5.11)$$

are simply the sums of the lengths of the edges along the paths from one leaf to the other leaf. These distances are intuitively what we would expect them to be. However, it is not clear intuitively whether the third distance,

$$d(1, 2) = -\ln\left(1 - e^{-\tau_3}\left(1 - e^{-2(\tau_1 + \tau_2)}\right)\right), \quad (5.12)$$

is an appropriate choice. To determine whether it is appropriate choice, we will look at its behaviour for various choices of  $\tau_3$ . Firstly, we will see what happens when  $\tau_3 = 0$ . When  $\tau_3 = 0$ , we simply have the three-taxon clock-like tree. The distance on the convergence-divergence network becomes

$$\begin{aligned} d(1, 2) &= -\ln\left(1 - e^{-0}\left(1 - e^{-2(\tau_1 + \tau_2)}\right)\right) \\ &= -\ln\left(1 - 1\left(1 - e^{-2(\tau_1 + \tau_2)}\right)\right) \\ &= -\ln\left(1 - 1 + e^{-2(\tau_1 + \tau_2)}\right) \\ &= -\ln\left(e^{-2(\tau_1 + \tau_2)}\right) \\ &= 2(\tau_1 + \tau_2), \end{aligned} \quad (5.13)$$

as is reasonable, since it is equal to the distance between the two equivalent leaves on the three-taxon clock-like tree.

Next, we will take the limit as  $\tau_3 \rightarrow \infty$  so that  $e^{-\tau_3} \rightarrow 0$ . The distance on the convergence-divergence satisfies

$$\begin{aligned} d(1, 2) &\rightarrow -\ln\left(1 - 0\left(1 - e^{-2(\tau_1 + \tau_2)}\right)\right) \\ &= -\ln(1) \\ &= 0. \end{aligned} \quad (5.14)$$

If taxon one and taxon two have converged for an infinitely long period of time the distance between the two taxa will be zero. This is consistent with what we expect with convergence.

For a “small” choice of  $\tau_3$ , the distance should be “slightly” less than when  $\tau_3 = 0$ . We will

let  $\tau_3 = \epsilon$ , a “small” positive number. When  $\tau_3 = \epsilon$ ,

$$\begin{aligned}
d(1, 2) &= -\ln \left( 1 - e^{-\epsilon} \left( 1 - e^{-2(\tau_1 + \tau_2)} \right) \right) \\
&\approx -\ln \left( 1 - (1 - \epsilon) \left( 1 - e^{-2(\tau_1 + \tau_2)} \right) \right) \\
&= -\ln \left( 1 - 1 + \epsilon + e^{-2(\tau_1 + \tau_2)} - \epsilon e^{-2(\tau_1 + \tau_2)} \right) \\
&= -\ln \left( e^{-2(\tau_1 + \tau_2)} + \epsilon \left( 1 - e^{-2(\tau_1 + \tau_2)} \right) \right) \\
&= -\ln \left( e^{-2(\tau_1 + \tau_2)} \left( 1 + \epsilon \frac{(1 - e^{-2(\tau_1 + \tau_2)})}{e^{-2(\tau_1 + \tau_2)}} \right) \right) \\
&= -\ln e^{-2(\tau_1 + \tau_2)} - \ln \left( 1 + \epsilon \frac{(1 - e^{-2(\tau_1 + \tau_2)})}{e^{-2(\tau_1 + \tau_2)}} \right) \\
&= 2(\tau_1 + \tau_2) - \ln \left( 1 + \epsilon \frac{(1 - e^{-2(\tau_1 + \tau_2)})}{e^{-2(\tau_1 + \tau_2)}} \right)
\end{aligned} \tag{5.15}$$

Before we expand the second term into its appropriate Taylor series, we need to know that the argument for the natural logarithm is in the interval  $(0, 2)$ ,

$$0 < 1 + \epsilon \frac{(1 - e^{-2(\tau_1 + \tau_2)})}{e^{-2(\tau_1 + \tau_2)}} < 2. \tag{5.16}$$

Since  $e^{-2(\tau_1 + \tau_2)} \in (0, 1]$  and  $1 - e^{-2(\tau_1 + \tau_2)} \in [0, 1)$ , it is clear that

$$1 + \epsilon \frac{(1 - e^{-2(\tau_1 + \tau_2)})}{e^{-2(\tau_1 + \tau_2)}} > 0. \tag{5.17}$$

For the other inequality,

$$\begin{aligned}
1 + \epsilon \frac{(1 - e^{-2(\tau_1 + \tau_2)})}{e^{-2(\tau_1 + \tau_2)}} &< 2 \\
\Leftrightarrow \epsilon \frac{(1 - e^{-2(\tau_1 + \tau_2)})}{e^{-2(\tau_1 + \tau_2)}} &< 1 \\
\Leftrightarrow \epsilon (1 - e^{-2(\tau_1 + \tau_2)}) &< e^{-2(\tau_1 + \tau_2)} \\
\Leftrightarrow \epsilon - \epsilon e^{-2(\tau_1 + \tau_2)} &< e^{-2(\tau_1 + \tau_2)} \\
\Leftrightarrow \epsilon &< (1 + \epsilon) e^{-2(\tau_1 + \tau_2)} \\
\Leftrightarrow \frac{\epsilon}{1 + \epsilon} &< e^{-2(\tau_1 + \tau_2)}.
\end{aligned} \tag{5.18}$$

Since we are taking  $\tau_1$  and  $\tau_2$  to be finite,  $e^{-2(\tau_1 + \tau_2)}$  will also be finite. As  $\epsilon \rightarrow 0$ ,  $\frac{\epsilon}{1 + \epsilon} \rightarrow 0$ . Therefore, there will always be some “small”  $\epsilon$  such that

$$\frac{\epsilon}{1 + \epsilon} < e^{-2(\tau_1 + \tau_2)}. \tag{5.19}$$

We will now go ahead and take the Taylor series expansion of the distance function,

$$\begin{aligned} d(1, 2) &\approx 2(\tau_1 + \tau_2) - \ln \left( 1 + \epsilon \frac{(1 - e^{-2(\tau_1 + \tau_2)})}{e^{-2(\tau_1 + \tau_2)}} \right) \\ &= 2(\tau_1 + \tau_2) - \epsilon \frac{(1 - e^{-2(\tau_1 + \tau_2)})}{e^{-2(\tau_1 + \tau_2)}} + O(\epsilon^2), \end{aligned} \quad (5.20)$$

where  $O(\epsilon^2)$  is all the terms of order  $\epsilon^2$  or higher.

If  $\epsilon$  is “small” then the lower order terms will dominate,

$$d(1, 2) \approx 2(\tau_1 + \tau_2) - \epsilon \frac{(1 - e^{-2(\tau_1 + \tau_2)})}{e^{-2(\tau_1 + \tau_2)}} \leq 2(\tau_1 + \tau_2), \quad (5.21)$$

as we require.

Hence, when  $\tau_3$  is “small” the two taxa will be closer in distance than when  $\tau_3 = 0$  by a “small” amount.

The last criterion that we want to check is that  $d(1, 2)$  decreases monotonically as  $\tau_3$  increases from  $\tau_3 = 0$ . To check this, we will look at the partial derivative of  $d(1, 2)$  with respect to  $\tau_3$ . The partial derivative is

$$\begin{aligned} \frac{\partial d(1, 2)}{\partial \tau_3} &= \frac{\partial}{\partial \tau_3} \left( -\ln \left( 1 - e^{-\tau_3} \left( 1 - e^{-2(\tau_1 + \tau_2)} \right) \right) \right) \\ &= -\frac{e^{-\tau_3} (1 - e^{-2(\tau_1 + \tau_2)})}{1 - e^{-\tau_3} (1 - e^{-2(\tau_1 + \tau_2)})} \leq 0, \end{aligned} \quad (5.22)$$

as we require.

It is important to note that our function,  $d(1, 2)$ , is not the only function that could behave in the appropriate manner for a distance. For  $d(1, 2)$  to be an appropriate distance function, it is necessary, but potentially not sufficient, for it to meet the criteria above.

We will expand the definition of tree distance to include distances between two leaves on a convergence-divergence network. We will call this distance the *pairwise distance*.

**Definition 5.5.2 (Pairwise Distance).** *The **pairwise distance**,  $d(a, b)$ , between two leaves,  $a$  and  $b$ , on a convergence-divergence network is expressed as*

$$d(a, b) = -\ln(q_A),$$

where  $A = i_1 i_2 \dots i_n$ ,  $a, b \in \{1, 2, \dots, n\}$ ,  $a \neq b$ ,  $i_a = i_b = 1$  and  $i_k = 0$  for all  $k \in \{1, 2, \dots, n\} \setminus \{a, b\}$ .

If the convergence-divergence network has no convergence and is a tree, then the pairwise distance will be equivalent to the tree distance.

We will now go ahead and compare the distances on different taxon label permutations of the three-taxon convergence-divergence network.

## 5.6 The (12) Taxon Label Permutation

We will now examine the convergence-divergence network with convergence between non-sister taxa, but with the first and second taxon labels permuted, the (12) taxon label permutation, shown in Figure 5.6.

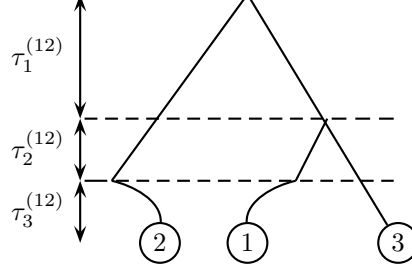


Figure 5.6: The (12) taxon label permutation on the three-taxon clock-like convergence-divergence network.

Recall from Chapter 4 that the constraints on the transformed phylogenetic tensor elements for the three-taxon clock-like convergence-divergence network were

$$\left\{ q_{011} \geq q_{101}, \quad q_{110} \geq q_{101}, \quad q_{011} (1 - q_{110})^2 \geq (q_{011} - q_{101})^2 \right\}. \quad (5.23)$$

On the (12) taxon label permutation the set of constraints becomes

$$\left\{ q_{101}^{(12)} \geq q_{011}^{(12)}, \quad q_{110}^{(12)} \geq q_{011}^{(12)}, \quad q_{101}^{(12)} (1 - q_{110}^{(12)})^2 \geq (q_{101}^{(12)} - q_{011}^{(12)})^2 \right\}. \quad (5.24)$$

If we compare this set of constraints to the set of constraints with no taxon label permutation, we find that the intersection of the two sets of constraints is empty, except for when  $q_{011} = q_{101}$ , where the two sets of constraints reduce to

$$\{q_{011} = q_{101}, \quad q_{110} \geq q_{101}\}. \quad (5.25)$$

This is the same set of constraints as the clock-like tree with the (13) and (123) taxon label permutations.

We can conclude that the clock-like convergence-divergence network with the (12) taxon label permutation is distinguishable from the clock-like convergence-divergence network with no taxon label permutation. We will not look at the distinguishability of other taxon label permutations from each other. Instead, we will examine whether they are network identifiable from each other.

Suppose now that we have a set of non-negative time parameters,  $\tau_1^{(0)}$ ,  $\tau_2^{(0)}$  and  $\tau_3^{(0)}$ , for the three-taxon clock-like convergence-divergence network with no taxon label permutation. Now suppose that we wish to find the time parameters on the (12) taxon label permutation that give rise to the same transformed phylogenetic tensor elements.



The expressions for the transformed phylogenetic tensor elements on the network with the (12) taxon label permutation in terms of the phylogenetic tensor elements on the network with no taxon label permutation are

$$\begin{cases} q_{101}^{(12)}(\tau_1^{(12)}, \tau_2^{(12)}, \tau_3^{(12)}) &= q_{011}^{(0)}(\tau_1^{(0)}, \tau_2^{(0)}, \tau_3^{(0)}), \\ q_{011}^{(12)}(\tau_1^{(12)}, \tau_2^{(12)}, \tau_3^{(12)}) &= q_{101}^{(0)}(\tau_1^{(0)}, \tau_2^{(0)}, \tau_3^{(0)}), \\ q_{110}^{(12)}(\tau_1^{(12)}, \tau_2^{(12)}, \tau_3^{(12)}) &= q_{110}^{(0)}(\tau_1^{(0)}, \tau_2^{(0)}, \tau_3^{(0)}). \end{cases} \quad (5.26)$$

We will now find expressions for the time parameters for the network with the (12) taxon label permutation in terms of the time parameters for the network with no taxon label permutation. The three pairwise distances for the network with no taxon label permutation are

$$\begin{cases} d(2,3) &= -\ln(q_{011}^{(0)}(\tau_1^{(0)}, \tau_2^{(0)}, \tau_3^{(0)})), \\ d(1,3) &= -\ln(q_{101}^{(0)}(\tau_1^{(0)}, \tau_2^{(0)}, \tau_3^{(0)})), \\ d(1,2) &= -\ln(q_{110}^{(0)}(\tau_1^{(0)}, \tau_2^{(0)}, \tau_3^{(0)})). \end{cases} \quad (5.27)$$

The three pairwise distances in terms of the transformed phylogenetic tensor elements for the (12) taxon label permutation are then

$$\begin{cases} d(2,3) &= -\ln(q_{101}^{(12)}(\tau_1^{(12)}, \tau_2^{(12)}, \tau_3^{(12)})), \\ d(1,3) &= -\ln(q_{011}^{(12)}(\tau_1^{(12)}, \tau_2^{(12)}, \tau_3^{(12)})), \\ d(1,2) &= -\ln(q_{110}^{(12)}(\tau_1^{(12)}, \tau_2^{(12)}, \tau_3^{(12)})). \end{cases} \quad (5.28)$$

We can now solve for the time parameters for the (12) taxon label permutation in terms of the time parameters for the network with no taxon label permutation by equating the two sets of expressions for the pairwise distances. For the derivation of the solutions to these equations see Appendix A. The time parameters for the network with the (12) taxon label permutation in

terms of the time parameters for the network with no taxon label permutation are

$$\begin{cases} \tau_1^{(12)} &= -\tau_1^{(0)}, \\ \tau_2^{(12)} &= \ln \left( \frac{\left(1 - e^{-2(\tau_1^{(0)} + \tau_2^{(0)})}\right) + \sqrt{\left(1 - e^{-2(\tau_1^{(0)} + \tau_2^{(0)})}\right)^2 + 4e^{-2\tau_2^{(0)}}}}{2e^{-(\tau_1^{(0)} + \tau_2^{(0)})}} \right), \\ \tau_3^{(12)} &= \ln \left( \frac{-\left(1 - e^{-2(\tau_1^{(0)} + \tau_2^{(0)})}\right) + \sqrt{\left(1 - e^{-2(\tau_1^{(0)} + \tau_2^{(0)})}\right)^2 + 4e^{-2\tau_2^{(0)}}}}{2e^{-(2\tau_2^{(0)} + \tau_3^{(0)})}} \right). \end{cases} \quad (5.29)$$

Clearly  $\tau_1^{(12)} \leq 0$ . We can conclude that the (12) taxon label permutation is network identifiable from no taxon label permutation. Although it is not necessary to look at  $\tau_2^{(12)}$  or  $\tau_3^{(12)}$ , we will for curiosity anyway. From Appendix A, we can conclude that  $\tau_2^{(12)}$  will always be non-negative, while  $\tau_3^{(12)}$  can be non-negative or negative. From simulations for  $x_1, x_2, x_3 \in U(0, 1)$ , we found that  $\tau_3^{(12)}$  was non-negative 434 times out of 500. We can conclude that the two taxon label permutations, the (12) taxon label permutation and no taxon label permutation, are network identifiable from each other.

The only case where both sets of time parameters are non-negative is when  $\tau_1^{(0)} = \tau_1^{(12)} = 0$ , where there is a trifurcation at the root of both of the convergence-divergence networks. When this happens all other time parameters for both taxon label permutations are non-negative, with  $\tau_2^{(0)} = \tau_2^{(12)}$  and  $\tau_3^{(0)} = \tau_3^{(12)}$ . If  $\tau_1^{(0)} > 0$  then it will not be possible to find a set of non-negative time parameters on the (12) taxon label permutation which gives rise to equivalent pairwise distances or transformed phylogenetic tensor elements on the convergence-divergence network with no taxon label permutation.

If we were given a pattern frequency and we wished to find the maximum likelihood for both of the taxon label permutations on the three-taxon convergence-divergence network then we would find that the maximum likelihoods would not be equal unless  $\tau_1^{(0)} = \tau_1^{(12)} = 0$ . Suppose  $\tau_1^{(0)}, \tau_2^{(0)}$  and  $\tau_3^{(0)}$  are all non-negative and are the maximum likelihood time parameters. For the maximum likelihood to be equal under the (12) taxon label permutation  $\tau_1^{(12)}$  cannot be positive and  $\tau_3^{(12)}$  may also be negative. If we constrain  $\tau_1^{(12)}, \tau_2^{(12)}$  and  $\tau_3^{(12)}$  to all be non-negative then the maximum likelihoods for both taxon label permutations will not be equal.

## 5.7 The (13) Taxon Label Permutation

We will now address the question of network identifiability of the (13) taxon label permutation from no taxon label permutation on the three-taxon clock-like convergence-divergence network. The (13) taxon label permutation on the three-taxon clock-like convergence-divergence network

is shown below in Figure 5.7.

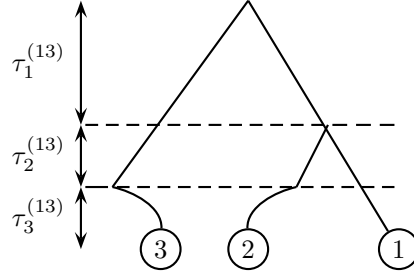


Figure 5.7: The (13) taxon label permutation on the three-taxon clock-like convergence-divergence network.

Equating the transformed phylogenetic tensor elements for this taxon label permutation to no taxon label permutation,

$$\begin{cases} q_{110}^{(13)}(\tau_1^{(13)}, \tau_2^{(13)}, \tau_3^{(13)}) &= q_{011}^{(1)}(\tau_1^{(1)}, \tau_2^{(1)}, \tau_3^{(1)}), \\ q_{101}^{(13)}(\tau_1^{(13)}, \tau_2^{(13)}, \tau_3^{(13)}) &= q_{101}^{(1)}(\tau_1^{(1)}, \tau_2^{(1)}, \tau_3^{(1)}), \\ q_{011}^{(13)}(\tau_1^{(13)}, \tau_2^{(13)}, \tau_3^{(13)}) &= q_{110}^{(1)}(\tau_1^{(1)}, \tau_2^{(1)}, \tau_3^{(1)}). \end{cases} \quad (5.30)$$

The three pairwise distances in terms of the transformed phylogenetic tensor elements for the (13) taxon label permutation are then

$$\begin{cases} d(2, 3) &= -\ln\left(q_{110}^{(13)}(\tau_1^{(13)}, \tau_2^{(13)}, \tau_3^{(13)})\right), \\ d(1, 3) &= -\ln\left(q_{101}^{(13)}(\tau_1^{(13)}, \tau_2^{(13)}, \tau_3^{(13)})\right), \\ d(1, 2) &= -\ln\left(q_{011}^{(13)}(\tau_1^{(13)}, \tau_2^{(13)}, \tau_3^{(13)})\right). \end{cases} \quad (5.31)$$

In Appendix B we solve for the time parameters for the (13) taxon label permutation in terms

of no taxon label permutation. Those time parameters are

$$\begin{cases} \tau_1^{(13)} = \ln \sqrt{\frac{1 - e^{-\tau_3^{(0)}} \left(1 - e^{-2(\tau_1^{(0)} + \tau_2^{(0)})}\right)}{e^{-2(\tau_1^{(0)} + \tau_2^{(0)} + \tau_3^{(0)})}}}, \\ \tau_2^{(13)} = \ln \left( \frac{\left(1 - e^{-2(\tau_2^{(0)} + \tau_3^{(0)})}\right) + \sqrt{\left(1 - e^{-2(\tau_2^{(0)} + \tau_3^{(0)})}\right)^2 + 4e^{-2(\tau_1^{(0)} + \tau_2^{(0)} + \tau_3^{(0)})}}}{2\sqrt{1 - e^{-\tau_3^{(0)}} \left(1 - e^{-2(\tau_1^{(0)} + \tau_2^{(0)})}\right)}} \right), \\ \tau_3^{(13)} = \ln \left( \frac{-\left(1 - e^{-2(\tau_2^{(0)} + \tau_3^{(0)})}\right) + \sqrt{\left(1 - e^{-2(\tau_2^{(0)} + \tau_3^{(0)})}\right)^2 + 4e^{-2(\tau_1^{(0)} + \tau_2^{(0)} + \tau_3^{(0)})}}}{2e^{-2(\tau_1^{(0)} + \tau_2^{(0)} + \tau_3^{(0)})}} \right). \end{cases} \quad (5.32)$$

In Appendix B we have shown that two of the time parameters on the (13) taxon label permutation,  $\tau_1^{(13)}$  and  $\tau_3^{(13)}$ , are non-negative, however we have also shown that  $\tau_2^{(13)}$  is almost always non-negative but can occasionally be negative. We can conclude that the (13) taxon label permutation is network identifiable from no taxon label permutation. However, for 497 out of 500 simulations for  $x_1, x_2, x_3 \in U(0, 1)$  all three of the time parameters were non-negative. In these cases the maximum likelihoods for the two taxon label permutations will be identical and we will have a choice between no taxon label permutation or the (13) taxon label permutation. We will discuss at the end of the chapter how we could choose between the two taxon label permutations.

## 5.8 The (23) Taxon Label Permutation

We will now address the question of network identifiability of the (23) taxon label permutation from no taxon label permutation on the three-taxon clock-like convergence-divergence network. The (23) taxon label permutation on the three-taxon clock-like convergence-divergence network is shown below in Figure 5.8.

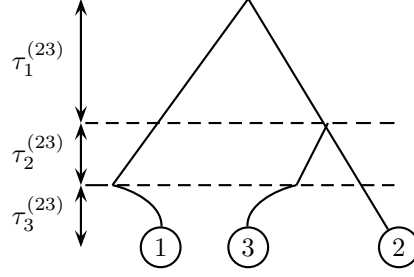


Figure 5.8: The (23) taxon label permutation on the three-taxon clock-like convergence-divergence network.

Equating the transformed phylogenetic tensor elements for this taxon label permutation to no taxon label permutation,

$$\begin{cases} q_{011}^{(23)}(\tau_1^{(23)}, \tau_2^{(23)}, \tau_3^{(23)}) = q_{011}^{(\cdot)}(\tau_1^{(\cdot)}, \tau_2^{(\cdot)}, \tau_3^{(\cdot)}), \\ q_{110}^{(23)}(\tau_1^{(23)}, \tau_2^{(23)}, \tau_3^{(23)}) = q_{101}^{(\cdot)}(\tau_1^{(\cdot)}, \tau_2^{(\cdot)}, \tau_3^{(\cdot)}), \\ q_{101}^{(23)}(\tau_1^{(23)}, \tau_2^{(23)}, \tau_3^{(23)}) = q_{110}^{(\cdot)}(\tau_1^{(\cdot)}, \tau_2^{(\cdot)}, \tau_3^{(\cdot)}). \end{cases} \quad (5.33)$$

The three pairwise distances in terms of the transformed phylogenetic tensor elements for the (23) taxon label permutation are then

$$\begin{cases} d(2, 3) = -\ln(q_{011}^{(23)}(\tau_1^{(23)}, \tau_2^{(23)}, \tau_3^{(23)})), \\ d(1, 3) = -\ln(q_{110}^{(23)}(\tau_1^{(23)}, \tau_2^{(23)}, \tau_3^{(23)})), \\ d(1, 2) = -\ln(q_{101}^{(23)}(\tau_1^{(23)}, \tau_2^{(23)}, \tau_3^{(23)})). \end{cases} \quad (5.34)$$

In Appendix C we solve for the time parameters for the (23) taxon label permutation in terms of no taxon label permutation. Those time parameters are

$$\begin{cases} \tau_1^{(23)} = \ln \sqrt{\frac{e^{-2(\tau_2^{(\cdot)} + \tau_3^{(\cdot)})}}{1 - e^{-\tau_3^{(\cdot)}}(1 - e^{-2(\tau_1^{(\cdot)} + \tau_2^{(\cdot)})})}}, \\ \tau_2^{(23)} = \ln \left( \frac{\left( (1 - e^{-2(\tau_1^{(\cdot)} + \tau_2^{(\cdot)} + \tau_3^{(\cdot)})}) + \sqrt{(1 - e^{-2(\tau_1^{(\cdot)} + \tau_2^{(\cdot)} + \tau_3^{(\cdot)})})^2 + 4(1 - e^{-\tau_3^{(\cdot)}}(1 - e^{-2(\tau_1^{(\cdot)} + \tau_2^{(\cdot)})})} \right)}{2e^{-(\tau_2^{(\cdot)} + \tau_3^{(\cdot)})}} \right), \\ \tau_3^{(23)} = \ln \left( \frac{-\left( (1 - e^{-2(\tau_1^{(\cdot)} + \tau_2^{(\cdot)} + \tau_3^{(\cdot)})}) + \sqrt{(1 - e^{-2(\tau_1^{(\cdot)} + \tau_2^{(\cdot)} + \tau_3^{(\cdot)})})^2 + 4(1 - e^{-\tau_3^{(\cdot)}}(1 - e^{-2(\tau_1^{(\cdot)} + \tau_2^{(\cdot)})})} \right)}{2(1 - e^{-\tau_3^{(\cdot)}}(1 - e^{-2(\tau_1^{(\cdot)} + \tau_2^{(\cdot)})})} \right). \end{cases} \quad (5.35)$$

From Appendix C, we have shown that  $\tau_1^{(23)}$  can be either non-negative or negative,  $\tau_2^{(23)}$

must be non-negative and  $\tau_3^{(23)}$  must be non-positive. When simulating samples for  $x_1, x_2, x_3 \in U(0, 1)$ , we found that  $\tau_1^{(23)}$  was non-negative 52 times out of 500. We can conclude that the (23) taxon label permutation is network identifiable from no taxon label permutation.

## 5.9 Comparing the (12) Permutation to the (23) Permutation

Having compared the (12) taxon label permutation to no taxon label permutation and the (23) taxon label permutation to no taxon label permutation, we will now compare the (12) taxon label permutation to the (23) taxon label permutation. We will address the question of network identifiability of the (23) taxon label permutation from the (12) taxon label permutation on the three-taxon clock-like convergence-divergence network. The (23) taxon label permutation on the three-taxon clock-like convergence-divergence network is shown in Figure 5.8 on page 111. The (12) taxon label permutation on the three-taxon clock-like convergence-divergence network is shown in Figure 5.6 on page 106.

Equating the transformed phylogenetic tensor elements for the two taxon label permutations,

$$\begin{cases} q_{011}^{(23)}(\tau_1^{(23)}, \tau_2^{(23)}, \tau_3^{(23)}) &= q_{101}^{(12)}(\tau_1^{(12)}, \tau_2^{(12)}, \tau_3^{(12)}), \\ q_{110}^{(23)}(\tau_1^{(23)}, \tau_2^{(23)}, \tau_3^{(23)}) &= q_{011}^{(12)}(\tau_1^{(12)}, \tau_2^{(12)}, \tau_3^{(12)}), \\ q_{101}^{(23)}(\tau_1^{(23)}, \tau_2^{(23)}, \tau_3^{(23)}) &= q_{110}^{(12)}(\tau_1^{(12)}, \tau_2^{(12)}, \tau_3^{(12)}). \end{cases} \quad (5.36)$$

The three pairwise distances in terms of the transformed phylogenetic tensor elements for the two taxon label permutations are then

$$\begin{cases} d(2, 3) = -\ln(q_{011}^{(23)}(\tau_1^{(23)}, \tau_2^{(23)}, \tau_3^{(23)})) &= -\ln(q_{101}^{(12)}(\tau_1^{(12)}, \tau_2^{(12)}, \tau_3^{(12)})), \\ d(1, 3) = -\ln(q_{110}^{(23)}(\tau_1^{(23)}, \tau_2^{(23)}, \tau_3^{(23)})) &= -\ln(q_{011}^{(12)}(\tau_1^{(12)}, \tau_2^{(12)}, \tau_3^{(12)})), \\ d(1, 2) = -\ln(q_{101}^{(23)}(\tau_1^{(23)}, \tau_2^{(23)}, \tau_3^{(23)})) &= -\ln(q_{110}^{(12)}(\tau_1^{(12)}, \tau_2^{(12)}, \tau_3^{(12)})). \end{cases} \quad (5.37)$$

In Appendix D we solve for the time parameters for the (23) taxon label permutation in terms

of the (12) label permutation. Those time parameters are

$$\begin{cases} \tau_1^{(23)} &= \ln \sqrt{\frac{e^{-2(\tau_1^{(12)} + \tau_2^{(12)} + \tau_3^{(12)})}}{1 - e^{-\tau_3^{(12)}} \left(1 - e^{-2(\tau_1^{(12)} + \tau_2^{(12)})}\right)}}, \\ \tau_2^{(23)} &= \ln \left( \frac{\left(1 - e^{-2(\tau_2^{(12)} + \tau_3^{(12)})}\right) + \sqrt{\left(1 - e^{-2(\tau_2^{(12)} + \tau_3^{(12)})}\right)^2 + 4 \left(1 - e^{-\tau_3^{(12)}} \left(1 - e^{-2(\tau_1^{(12)} + \tau_2^{(12)})}\right)\right)}}{2e^{-(\tau_1^{(12)} + \tau_2^{(12)} + \tau_3^{(12)})}} \right), \\ \tau_3^{(23)} &= \ln \left( \frac{-\left(1 - e^{-2(\tau_2^{(12)} + \tau_3^{(12)})}\right) + \sqrt{\left(1 - e^{-2(\tau_2^{(12)} + \tau_3^{(12)})}\right)^2 + 4 \left(1 - e^{-\tau_3^{(12)}} \left(1 - e^{-2(\tau_1^{(12)} + \tau_2^{(12)})}\right)\right)}}{2 \left(1 - e^{-\tau_3^{(12)}} \left(1 - e^{-2(\tau_1^{(12)} + \tau_2^{(12)})}\right)\right)} \right). \end{cases} \quad (5.38)$$

From Appendix D, we have shown that  $\tau_1^{(23)}$  must be non-positive,  $\tau_2^{(23)}$  must be non-negative and  $\tau_3^{(23)}$  can be non-negative or negative. When simulating samples for  $x_1, x_2, x_3 \in U(0, 1)$ , we found that  $\tau_3^{(23)}$  was non-negative 74 times out of 500. We can conclude that the (23) taxon label permutation is network identifiable from the (12) taxon label permutation.

## 5.10 The Six Taxon Label Permutations

Recall that there are six taxon label permutations, described by the symmetric group,

$$S_3 = \{(), (12), (13), (23), (123), (132)\}. \quad (5.39)$$

We can express these group elements as products of other group elements. For example, we can express every group element as a product of another group element and (13),

$$\begin{cases} () &= (13) \cdot (13), \\ (12) &= (13) \cdot (123), \\ (13) &= (13) \cdot (), \\ (23) &= (13) \cdot (132), \\ (132) &= (13) \cdot (23), \\ (123) &= (13) \cdot (12). \end{cases} \quad (5.40)$$

Recall that when the time parameters on no taxon label permutation are non-negative, the time parameters on the (13) taxon label permutation are almost always non-negative as well. When we compare the two taxon label permutations we see that they have the same pairwise distance,  $d(1, 3)$ . This pairwise distance is twice the sum of the time parameters on the network. However, for the pairwise distances to be identical on both taxon label permutations the second time parameter on the (13) taxon label permutation,  $\tau_2^{(12)}$ , must occasionally be negative.

From a symmetry argument, we can conclude that when the time parameters for the (13) taxon label permutation in terms of no taxon label permutation are all non-negative, the time parameters for the (123) taxon label permutation in terms of the (12) taxon label permutation must all be non-negative and the time parameters for the (132) taxon label permutation in terms of the (23) taxon label permutation must all be non-negative also. In these cases there will be three pairs of identical maximum likelihoods: no taxon label permutation and the (13) taxon label permutation, the (12) taxon label permutation and the (123) taxon label permutation, the (23) taxon label permutation and the (132) taxon label permutation.

In these circumstances we will have a choice between two taxon label permutations that both maximise the likelihood. The time parameters, however, will be different. If it is known or suspected that two taxa have converged then we may wish to choose the taxon label permutation that allows for this convergence. We may also choose the taxon label permutation that allows for the shortest convergence time or we may choose the taxon label permutation that converges the two leaves that have the smallest pairwise distance before convergence.



## Chapter 6

# Distinguishability of Trees and Networks using Gröbner Bases

In Chapter 4 we looked at the identifiability and distinguishability of trees and networks for two and three taxa. Determining the distinguishability of two-taxon and three-taxon trees and networks was fairly straightforward. To address the issue of distinguishability we compared the constraints on the phylogenetic tensors for various trees and networks, as well as taxon label permutations. For trees and networks with more than three taxa or with a large number of convergence periods we will need a more sophisticated method for determining distinguishability.

Recall that the basis of the phylogenetic tensors was first transformed. The phylogenetic tensor elements, both in the original and transformed bases, are expressions in terms of the time parameters on the tree or network. The issue of identifiability was addressed by finding expressions for the time parameters on the trees and networks in terms of the transformed phylogenetic tensor elements. If expressions for every time parameter were able to be found independently of every other time parameter then the tree or network was identifiable. For both two and three taxa, the equality and inequality constraints on the elements of the transformed phylogenetic tensor were then found and compared between various trees and networks to address the issue of distinguishability between the trees and networks.

For trees and networks with large numbers of taxa the phylogenetic tensors are of greater dimension with more time parameters, corresponding to a system with more equations and more time parameters to solve for. For the binary symmetric model, the phylogenetic tensors are of dimension  $2^n$ , where  $n$  is the number of taxa. It is easy to see that for large  $n$  the system will contain a large number of equations and solving the system will become progressively more challenging as  $n$  becomes larger. To compare trees and networks with a large number of taxa it will be necessary to employ a more efficient technique for dealing with our large systems of equations.

Recall that the systems of equations for the transformed phylogenetic tensor elements from

our two-taxon and three-taxon examples are non-linear. Since the systems are non-linear we cannot simply use Gaussian elimination to solve them. Likewise, the systems of equations for the transformed phylogenetic tensor elements for four-taxon trees and networks, and more generally  $n$ -taxon trees and networks, will be non-linear. Our equations involve exponential quantities, with the arguments of these exponential quantities being the negative of a sum of time parameters. The arguments of the exponential quantities are therefore non-positive. We made the simple substitutions,  $y_i = e^{-\tau_i}$ , where  $\tau_i$  is a dimensionless time parameter dependent on the product of two scalars, the positive rate parameter from the binary symmetric model and a non-negative time parameter from the tree or network. Since the units for the rate parameters are the inverse of time,  $\tau_i$  will be a non-negative real number with no units. We saw in Chapter 4 and Chapter 5 that the systems of equations become systems of polynomial equations in the variables  $y_i$  after these substitutions are made. From here, we can use techniques from algebraic geometry to solve the systems and find their constraints. We will see how this is done by looking at a three-taxon example that we have already examined in Section 4.3.2 before addressing several four-taxon examples. We will show that the technique is useful for networks with many taxa or many convergence periods.

Before we solve the systems of polynomials which define the phylogenetic tensors, we must first introduce the basic concepts we require from algebraic geometry. An excellent source to introduce readers to the algebraic geometric concepts discussed here, including the proofs of all of the theorems we require, is Little et al. [1992]. The algebraic geometric concepts that we will discuss, including the notation, will be similar to that used by Little et al. [1992].

The example we will use to illustrate the processes involved will be the three-taxon clock-like tree, shown below in Figure 6.1.

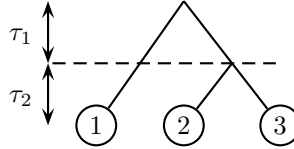


Figure 6.1: Three-taxon clock-like tree.

The transformed phylogenetic tensor for the three-taxon clock-like tree is

$$\hat{P} = \begin{bmatrix} q_{000} \\ q_{001} \\ q_{010} \\ q_{011} \\ q_{100} \\ q_{101} \\ q_{110} \\ q_{111} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ e^{-2\tau_2} \\ 0 \\ e^{-2(\tau_1+\tau_2)} \\ e^{-2(\tau_1+\tau_2)} \\ 0 \end{bmatrix}. \quad (6.1)$$

We have already found the constraints for this tree, however this is a simple example to introduce to the reader the techniques in algebraic geometry required to find the equality and inequality constraints on the transformed phylogenetic tensor.

## 6.1 Algebraic Geometric Techniques for Finding the Constraints

Suppose we are given a set of linear equations. Provided the system is not underdetermined then it can be solved through Gaussian elimination. However, if we are given a set of non-linear equations then there is no general algorithm to solve them unless they are polynomial equations. We have seen that for every two-taxon and three-taxon tree or network the equations for the transformed phylogenetic tensor elements could be converted into polynomial equations. We will see that the equations for the transformed phylogenetic tensor elements for every four-taxon tree or network can also be converted into polynomial equations. We will discuss the relevant techniques in algebraic geometry required to solve polynomial equations and apply these techniques to the example we have introduced before examining some four-taxon trees and networks.

We will begin by formally defining monomials and polynomials. We will follow this with a discussion on how the terms of a polynomial in one variable can be ordered by their degrees. Next, we will show how the ordering of polynomials is used in the division algorithm for polynomial division. We will then discuss how multi-variable monomials can be ordered when there is no immediately obvious generalisation from the single variable ordering by degrees.

The next step will be to define ideals, subrings that can be used to solve systems of polynomial equations. Given a system of polynomial equations,  $f_1, f_2, \dots, f_s = 0$ , we can make the polynomials,  $f_1, f_2, \dots, f_s$ , a generating set of polynomials for an ideal. The set of polynomial generators of a polynomial ideal is called the basis and is not unique.

We will then introduce two specific types of monomial orderings: lex ordering and product ordering. Lex ordering is analogous to the ordering of words in the dictionary and will be applied to each of the “types” of variable: the time variables and the transformed phylogenetic tensor elements. The product ordering allows us to partition the two different types of variables.

We will mention Buchberger’s algorithm, which uses a multi-variable generalisation of the division algorithm to find the reduced Gröbner basis of the ideal. By choosing the reduced Gröbner basis, the remainder from the division algorithm will always be unique, a property which is not generally true of the division algorithm in multiple variables. The set of polynomials in the reduced Gröbner basis, or any basis for that matter, generates the same ideal that the original set of polynomials generated.

We can find the solutions to the system of polynomial equations from the set of polynomials in the reduced Gröbner basis of the ideal. By using a lex monomial ordering we can eliminate the variables successively in a fashion analogous to back-substitution in Gaussian elimination.

By using the product order we can separate the two types of variables, as mentioned earlier. The generators in the reduced Gröbner basis will then be a mix of two “types” of polynomials: the first containing only the transformed phylogenetic tensor elements and the second containing both the transformed phylogenetic tensor elements and the time variables.

The equality constraints on the transformed phylogenetic tensor elements can be found from the polynomials of the first type. From the second type of polynomials we can find solutions for the time variables in terms of the transformed phylogenetic tensor elements. Recall that the time variables involved exponential quantities and were of the form  $y_i = e^{-\tau_i}$ , where  $\tau_i$  is a dimensionless time parameter. We will demand the time variables be constrained to  $0 < y_i \leq 1$ . Demanding these constraints on the time variables will then give us a set of inequality constraints on the transformed phylogenetic tensor elements.

To avoid confusion, we let  $x_1, x_2, \dots, x_m$  be the variables in our definitions and let  $y_1, y_2, \dots, y_m$  be the variables for our examples.

We will start by defining a *monomial*.

**Definition 6.1.1 (Monomial).** *In the scalar variables,  $x_1, x_2, \dots, x_m$ , a **monomial** is a single term,*

$$x_1^{d_1} \cdot x_2^{d_2} \cdot \dots \cdot x_m^{d_m},$$

where  $d_i \in \mathbb{N}_{\geq 0}$  for all  $1 \leq i \leq m$  and  $\mathbb{N}_{\geq 0} = \{0, 1, \dots\}$ .

We can express the monomial as

$$x^d = x_1^{d_1} \cdot x_2^{d_2} \cdot \dots \cdot x_m^{d_m}, \quad (6.2)$$

where  $d \in \mathbb{N}_{\geq 0}^m$ .

**Definition 6.1.2 (Total degree).** *The **total degree** of the monomial,  $|d|$ , is*

$$|d| = \sum_{i=1}^m d_i.$$

For example, we make the substitution,  $y_2 = e^{-\tau_2}$ . Thus  $e^{-2\tau_2} \mapsto y_2^2$  and is now a monomial. Having defined a monomial, we are now in a position to define a *polynomial*.

**Definition 6.1.3 (Polynomial).** *A **polynomial**,  $f$ , is a finite linear sum of monomials, with the coefficients of the monomials belonging to some arbitrary field,  $\mathbb{F}$ . In terms of monomials,*

$$f = \sum_d a_d x^d,$$

where  $a_d \in \mathbb{F}$ .

The polynomial ring  $\mathbb{F}[x_1, x_2, \dots, x_m]$  is formed from the set of all polynomials in  $x_1, x_2, \dots, x_m$ , with the coefficients belonging to  $\mathbb{F}$ .

Referring to our example, we can now see that if  $q_{011} \in \mathbb{F}$  then  $y_2^2 - q_{011} = 0$  is an equation with  $y_2^2 - q_{011}$  being a polynomial in the ring,  $\mathbb{F}[y_2]$ . We will now discuss some definitions relating to polynomials before introducing the notion of an *ordering* of monomials, which is required for our process.

**Definition 6.1.4.** Given the polynomial,  $f = \sum_d a_d x^d$ , in  $\mathbb{F}[x_1, x_2, \dots, x_m]$ ,

- (i)  $a_d$  is the **coefficient** of  $x^d$ ,
- (ii)  $a_d x^d$  is a **term** of  $f$ , provided  $a_d \neq 0$ ,
- (iii) The **total degree** of  $f$  is  $\deg(f) = \max(|d|)$ , where  $a_d \neq 0$ .

We can order the polynomial terms based on the degrees of those terms. For example, we can order an arbitrary polynomial in one variable,  $x$ , (or any number of variables, for that matter) in decreasing order, from the term with the highest degree first to the term with the lowest degree last. An arbitrary polynomial in one variable can be written in the form

$$f = a_1 x^p + a_2 x^{p-1} + \dots + a_p x + a_{p+1}, \quad (6.3)$$

where  $a_i \in \mathbb{F}$ ,  $a_1 \neq 0$ ,  $p \in \mathbb{N}_{\geq 0}$  and some coefficients may be zero.

**Definition 6.1.5 (Leading Term).** The term with the highest degree, or the total degree of the polynomial,  $LT(f) = a_1 x^p$ , is called the **leading term**.

We can now define polynomial division, which is necessary for finding the Gröbner basis of the ideal. By comparing the leading terms of two polynomials we can determine whether the leading term of one of those polynomials *divides* the leading term of the other polynomial. Suppose we have two polynomials,  $f_1$  and  $f_2$ , in one variable,  $x$ .

**Definition 6.1.6 (Polynomial Division).** For two polynomials,  $f_1, f_2 \in \mathbb{F}[x]$ ,  $LT(f_2)$  divides  $LT(f_1)$  if and only if  $\deg(f_2) \leq \deg(f_1)$ .

If  $LT(f_2)$  divides  $LT(f_1)$ , then  $f_1$  can be expressed as

$$f_1 = s f_2 + r, \quad (6.4)$$

where  $r, s \in \mathbb{F}[x]$ ,  $\deg(r) < \deg(f_2)$  and possibly  $r = 0$ . If  $r = 0$  then not only does  $LT(f_2)$  divide  $LT(f_1)$ , but  $f_2$  divides  $f_1$ .

Given two polynomials,  $f_1$  and  $f_2$ , in one variable,  $x$ ,  $r$  and  $s$  will be unique. The *division algorithm* can be used to find  $r$  and  $s$ .

**Proposition 6.1.7 (Division Algorithm).** Start with  $s = 0$  and  $r = f_1$ .

**while**  $r \neq 0$  **and**  $LT(f_2)$  **divides**  $LT(r)$  **do**  
 $s \leftarrow s + \frac{LT(r)}{LT(f_2)}$   
 $r \leftarrow r - \left(\frac{LT(r)}{LT(f_2)}\right) f_2$   
**end while**

We will now look at a simple example to illustrate how one polynomial can divide another. Suppose we wanted to divide  $f_1 = x^2 - 1$  by  $f_2 = x + 1$  and express  $f_1$  as  $f_1 = sf_2 + r$ . We start by setting  $s = 0$  and  $r = f_1 = x^2 - 1$ . We now check that  $r = x^2 - 1 \neq 0$  and  $\deg(LT(f_2)) = \deg(x) = 1 \leq \deg(LT(r)) = \deg(x^2) = 2$ . Since both are true,  $s$  and  $r$  become

$$\begin{aligned} s &\leftarrow 0 + \frac{x^2}{x} = x, \\ r &\leftarrow x^2 - 1 - \frac{x^2}{x}(x + 1) = -x - 1. \end{aligned} \tag{6.5}$$

We again check that  $r = -x - 1 \neq 0$  and  $\deg(LT(f_2)) = \deg(x) = 1 \leq \deg(LT(r)) = \deg(-x) = 1$ . Since the coefficient of the leading term has no influence on its degree, both statements are again true.  $s$  and  $r$  then become

$$\begin{aligned} s &\leftarrow x + \frac{-x}{x} = x - 1, \\ r &\leftarrow -x - 1 - \frac{-x}{x}(x + 1) = 0. \end{aligned} \tag{6.6}$$

Since  $r = 0$  the algorithm terminates and we have successfully divided  $f_1 = x^2 - 1$  by  $f_2 = x + 1$ .  $f_1$  can now be expressed as,

$$f_1 = sf_2 + r, \tag{6.7}$$

where  $r = 0$  and  $s = x - 1$ .

It can be seen that the division algorithm in  $\mathbb{F}[x]$  and row-reduction algorithm for linear systems (Gaussian elimination with matrices) depend on an *ordering of terms* of polynomials. The polynomials can be ordered by total degree. For our multi-variable scenario it is necessary to introduce the notion of *monomial ordering*.

**Definition 6.1.8 (Monomial Ordering).** Suppose we have the monomials,  $x^d$ ,  $x^e$  and  $x^g$ , where  $d, e, g \in \mathbb{N}_{\geq 0}^m$ . A **monomial ordering** is an ordering,  $>$ , on the monomials, which meets

the following conditions:

- (i)  $x^d \leq x^e$  and  $x^e \leq x^d \Rightarrow x^d = x^e$ . This is the antisymmetry condition,
- (ii)  $x^d \leq x^e$  and  $x^e \leq x^g \Rightarrow x^d \leq x^g$ . This is the transitivity condition,
- (iii)  $x^d \leq x^e$  or  $x^e \leq x^d$ . This is the totality condition,
- (iv)  $x^d > x^e$  and  $g \in \mathbb{N}_{\geq 0}^m \Rightarrow x^d x^g > x^e x^g$ ,
- (v) There is a minimum monomial for every non-empty subset of  $\mathbb{N}_{\geq 0}^m$  under  $>$ ,  
with the ordering  $0 < 1 < \dots$ .

Conditions (i) to (iii) are the restriction that  $>$  must be a total (or linear) ordering on  $\mathbb{N}_{\geq 0}^m$ . Condition (iv) says that the ordering of two monomials is only dependent on the variables that differ in their degrees. Condition (v) says that  $>$  must be a well-ordering on  $\mathbb{N}_{\geq 0}^m$ , a technical point not needed to understand our general procedure.

Suppose we have three monomials,  $x^a$ ,  $x^b$  and  $x^c$ , where  $a, b, c \in \mathbb{N}_{\geq 0}^3$ , where the three monomials are

$$\begin{cases} x^a = x_1^{a_1} x_2^{a_2} x_3^{a_3} &= x_1^2 x_2^5 x_3, \\ x^b = x_1^{b_1} x_2^{b_2} x_3^{b_3} &= x_1^3 x_3, \\ x^c = x_1^{c_1} x_2^{c_2} x_3^{c_3} &= x_1^3 x_2^2 x_3. \end{cases} \quad (6.8)$$

An example of a monomial ordering is lex ordering, which will be discussed in more detail later. Under lex ordering of monomials, the degrees of each variable in each monomial are compared successively. Comparing two monomials, the degrees of the first variable are compared first. If these degrees are equal, then the degrees of the second variable are then compared. The process is continued until the two monomials differ in their degrees for a given variable. If the two monomials never differ in their degrees then they are identical and equal under any monomial ordering. Comparing  $x^a$ ,  $x^b$  and  $x^c$ ,  $b_1 = c_1 = 3 > a_1 = 2$ . We can conclude that  $x^b, x^c > x^a$ . To determine whether  $x^b > x^c$ ,  $x^b = x^c$  or  $x^b < x^c$ , we must look at the second variable. Clearly,  $c_2 = 2 > b_2 = 0$  and  $x^c > x^b$ . In conclusion,  $x^c = x_1^3 x_2^2 x_3 > x^b = x_1^3 x_3 > x^a = x_1^2 x_2^5 x_3$  under lex ordering.

**Definition 6.1.9.** Suppose we have a nonzero polynomial in  $\mathbb{F}[x_1, x_2, \dots, x_m]$ ,  $f = \sum_d a_d x^d$ , with the monomial order,  $>$ .

- (i)  $\text{multideg}(f) = \max(d \in \mathbb{N}_{\geq 0}^m : a_d \neq 0)$  is the **multidegree** of  $f$ ,
- (ii)  $LC(f) = a_{\text{multideg}(f)} \in \mathbb{F}$  is the **leading coefficient** of  $f$ ,
- (iii)  $LM(f) = x^{\text{multideg}(f)}$  is the **leading monomial** of  $f$ ,
- (iv)  $LT(f) = LC(f) \cdot LM(f)$  is the **leading term** of  $f$ .

In order to solve a system of polynomial equations,  $f_1, f_2, \dots, f_s = 0$ , we will introduce an

ideal, which can be expressed in terms of a system of polynomial equations.

**Definition 6.1.10 (Ideal).** An **ideal** is a set,  $I \subset \mathbb{F}[x_1, x_2, \dots, x_m]$ , that satisfies the conditions:

- (i)  $0 \in I$ ,
- (ii) If  $f_1, f_2 \in I$  then  $f_1 + f_2 \in I$ ,
- (iii) If  $f_1 \in I$  and  $s \in \mathbb{F}[x_1, x_2, \dots, x_m]$  then  $sf_1 \in I$ .

An example of an ideal can be expressed in terms of a set of *generators*.

**Definition 6.1.11.** For any set of polynomials,  $f_1, f_2, \dots, f_s \in \mathbb{F}[x_1, x_2, \dots, x_m]$ , we define the set  $I = \langle f_1, f_2, \dots, f_s \rangle$ , as follows:

$$I = \langle f_1, f_2, \dots, f_s \rangle = \left\{ \sum_{i=1}^s h_i f_i : h_1, h_2, \dots, h_s \in \mathbb{F}[x_1, x_2, \dots, x_m] \right\},$$

where  $f_1, f_2, \dots, f_s$  are called the **generators** of the ideal. Different sets of generators are referred to as different **bases** for the ideal.

It is easy to check that  $I = \langle f_1, f_2, \dots, f_s \rangle$  is an ideal for *any* polynomials  $f_1, f_2, \dots, f_s \in \mathbb{F}[x_1, x_2, \dots, x_m]$ . The set of generators of an ideal is not unique. If we have multiple sets of generators for an ideal then the sets of polynomial equations that we form from the generators must all have the same solutions. Some sets of generators may, however, allow for the polynomial equations to be solved more easily.

After making the appropriate substitutions,  $y_i = e^{-\tau_i}$ , our equations for the transformed phylogenetic tensor elements are all of the form,

$$q_{i_1 i_2 \dots i_n} = f(y_1, y_2, \dots, y_m), \quad (6.9)$$

where  $q_{i_1 i_2 \dots i_n} \in \mathbb{F}$  and  $f(y_1, y_2, \dots, y_m) \in \mathbb{F}[y_1, y_2, \dots, y_m]$ .

The polynomial equations are then rearranged to be

$$f(y_1, y_2, \dots, y_m) - q_{i_1 i_2 \dots i_n} = 0. \quad (6.10)$$

We then take each  $f(y_1, y_2, \dots, y_m) - q_{i_1 i_2 \dots i_n}$  to be a generator in the generating set of polynomials for an ideal.

For the three-taxon clock-like tree, our set of generating polynomials is  $\{y_2^2 - q_{011}, y_1^2 y_2^2 - q_{101}, y_1^2 y_2^2 - q_{110}\}$ , which generates the ideal  $\langle y_2^2 - q_{011}, y_1^2 y_2^2 - q_{101}, y_1^2 y_2^2 - q_{110} \rangle$ .

We would now like to transform the basis of the ideal. By transforming the basis of the ideal, the new generators of the ideal will form an equivalent set of equations which may be able to be solved more easily. Likewise, any equality constraints involving the phylogenetic tensor elements can be found from the generators of the ideal if they are in the appropriate basis. We will now specify the monomial ordering necessary for our basis transformation process.



**Definition 6.1.12 (Lexicographic (Lex) Order).**  $x^d >_{\text{lex}} x^e$ , where  $d, e \in \mathbb{N}_{\geq 0}^m$ , if the first non-zero element in  $d - e = (d_1 - e_1, d_2 - e_2, \dots, d_m - e_m) \in \mathbb{N}_{\geq 0}^m$ , is greater than zero.

We will apply the lex order  $y_1 > y_2 > \dots > y_m$  to our  $m$  time variables, where  $y_i = e^{-\tau_i}$  for all  $1 \leq i \leq m$ . Lex order derives its name from an analogy to the ordering of words in a dictionary. For example, if we take the words “algorithm” and “arbitrary” and assign them the strings “ $x_1x_2\dots x_9$ ” and “ $x'_1x'_2\dots x'_9$ ” respectively, we see that  $x_1 = a =_{\text{lex}} x'_1 = a$  and  $x_2 = l >_{\text{lex}} x'_2 = r$ . Hence, *algorithm*  $>_{\text{lex}}$  *arbitrary*. If words are ordered in descending order according to this choice of lex ordering then it is equivalent to the reverse alphabetical order. Two more examples of lex order in use are  $x_1x_2x_3 >_{\text{lex}} x_2x_3$  and  $x_1^2x_2x_3 >_{\text{lex}} x_1x_2x_3^2$ .

**Definition 6.1.13 (Block (or Elimination) Order).** A **block order** is a monomial order on  $\mathbb{F}[x_1, x_2, \dots, x_m, x'_1, x'_2, \dots, x'_p]$  for  $x_1, x_2, \dots, x_m$  if every polynomial which has its leading monomial in  $\mathbb{F}[x'_1, x'_2, \dots, x'_p]$  is inside  $\mathbb{F}[x'_1, x'_2, \dots, x'_p]$ . In other words,

$$LM(g) \in \mathbb{F}[x'_1, x'_2, \dots, x'_p] \Rightarrow g \in \mathbb{F}[x'_1, x'_2, \dots, x'_p].$$

For more on block ordering and its relevance to the Elimination Theorem, a theorem necessary to our process that will not be discussed further, see [Hassett, 2007]. For our variables, we will set  $y_1, y_2, \dots, y_m$  as the  $m$  time variables, as before. The remaining variables will be the phylogenetic tensor elements,  $q_{i_1i_2\dots i_n}$ . We will use lex order for the two sets of variables, the  $m$  time variables and the variables for the phylogenetic tensor elements.

**Definition 6.1.14 (Product Order).** The product order,  $>_{\text{prod}}$ , is defined as

$$x^d x'^e >_{\text{prod}} x^g x'^h \text{ if } \begin{cases} x^d >_{\text{lex}} x^g, \\ \text{or } x^d =_{\text{lex}} x^g \text{ and } x'^e >_{\text{lex}} x'^h. \end{cases}$$

A product order is a type of block order. The product ordering will allow us to solve the system of equations for the time variables,  $y_1, y_2, \dots, y_m$ , in terms of the transformed phylogenetic tensor variables,  $q_{i_1i_2\dots i_n}$ , while retaining any equality constraints involving the transformed phylogenetic tensor variables.

Having chosen a monomial order for our variables, we will restrict our coefficients to belong to the rationals,  $\mathbb{Q}$ . We can now apply the multivariable division algorithm to attempt to find an ideal equivalent to our existing ideal, but in a different basis with different generators. This will allow our polynomial equations to be solved more easily and will give any equality constraints on the transformed phylogenetic tensor variables. Recall that the remainder for the division algorithm is unique in the one variable case. Unfortunately, in multiple variables the division algorithm is not sufficient to uniquely characterise the remainder,  $r$ , upon division by a generating set of polynomials. By choosing the *Gröbner basis* as the basis for our ideals,  $r$  is uniquely determined and  $r = 0$  is equivalent to the polynomial belonging to the ideal.

**Definition 6.1.15 (Gröbner (or Standard) Basis).** Suppose we are given an ideal,  $I = \langle f_1, f_2, \dots, f_s \rangle$ , and a monomial ordering. A finite subset,  $G = \{g_1, g_2, \dots, g_t\}$ , of the ideal, is a Gröbner (or standard) basis if

$$\langle LT(g_1), LT(g_2), \dots, LT(g_t) \rangle = \langle LT(I) \rangle = \langle LT(f_1), LT(f_2), \dots, LT(f_s) \rangle.$$

**Corollary 6.1.16.** For a given monomial order, a Gröbner basis exists for every ideal,  $I \subset \mathbb{F}[x_1, x_2, \dots, x_m]$ ,  $I \neq \{0\}$ .

*Proof.* Proof can be found in Little et al. [1992].  $\square$

If we are given a basis for an ideal, we can determine whether the basis is a Gröbner basis for the ideal. Following from the definition of an ideal, we can determine if a monomial lies in a monomial ideal.

**Lemma 6.1.17.** Suppose  $I = \langle x^d : d \in D \rangle$  is a monomial ideal, where  $D \subset \mathbb{N}_{\geq 0}^m$ . Then a given monomial,  $x^e$ , is in  $I$  if and only if there is a  $d \in D$  for which  $x^e$  is divisible by  $x^d$ .

*Proof.* Proof can be found in Little et al. [1992].  $\square$

**Corollary 6.1.18.** For a given monomial order, every ideal,  $I \subset \mathbb{F}[x_1, x_2, \dots, x_m]$ ,  $I \neq \{0\}$ , has a Gröbner basis and any Gröbner basis for an ideal,  $I$ , is a basis of  $I$ .

*Proof.* Proof can be found in Little et al. [1992].  $\square$

**Definition 6.1.19 (Reduced Gröbner Basis).** A reduced Gröbner basis is a Gröbner basis,  $G = \{g_1, g_2, \dots, g_t\}$ , for an ideal,  $I$ , which meets the requirements:

- (i)  $LC(p) = 1$  for all  $p \in G$ ,
- (ii) No monomial of  $p$  is in  $\langle LT(G - \{p\}) \rangle$  for all  $p \in G$ .

**Proposition 6.1.20.** Suppose  $I \neq \{0\}$  is an ideal. For a fixed monomial order,  $I$  will have a unique reduced Gröbner basis.

*Proof.* Proof can be found in Little et al. [1992].  $\square$

A Gröbner basis for a polynomial ideal,  $I \neq \{0\}$ , can be found using *Buchberger's algorithm*. Buchberger's algorithm uses the multivariable division algorithm to transform the basis of an ideal into the Gröbner basis. For more on Buchberger's algorithm and the proof that it produces a Gröbner basis, see Little et al. [1992]. We will use the computer algebra system *Macaulay2* to compute our Gröbner bases. Sturmfels [2002] explains how to use *Macaulay2* to compute a reduced Gröbner basis for a given monomial order.

We will choose the field to be the rationals,  $\mathbb{Q}$ , the variables to be  $y_1, y_2, \dots, y_m$  and  $q_{00\dots 00}, q_{00\dots 01}, q_{00\dots 10}, \dots, q_{11\dots 11}$ , the ring to be the set of all polynomials in our stated variables, with

coefficients belonging to our field. We will choose the lex ordering of  $y_1 > y_2 > \dots > y_m$ , the lex ordering of the transformed phylogenetic tensor variables of  $q_{00\dots 00} > q_{00\dots 01} > q_{00\dots 10} > \dots > q_{11\dots 11}$  and product ordering for  $y_1, y_2, \dots, y_m$ . The lex ordering on the transformed phylogenetic tensor variables is the ordering of the tensor elements determined by the Kronecker product.

Once we have found the generators of the Gröbner basis, the transformed phylogenetic tensor variables,  $q_{i_1 i_2 \dots i_n}$ , will then be interpreted as constants. This allows the polynomial equations to be solved for the variables,  $y_1, y_2, \dots, y_m$ , as functions of the transformed phylogenetic tensor variables,  $q_{i_1 i_2 \dots i_n}$ . Additionally, any constraints involving only the transformed phylogenetic tensor variables,  $q_{i_1 i_2 \dots i_n}$ , and not  $y_1, y_2, \dots, y_m$  will have also been found explicitly.

We will now refer back to our ideal for the three-taxon clock-like tree,

$$I = \langle y_2^2 - q_{011}, y_1^2 y_2^2 - q_{101}, y_1^2 y_2^2 - q_{110} \rangle. \quad (6.11)$$

For any of our trees or networks, we can use the following *Macaulay2* code to find the Gröbner basis of the ideal.

—Field of rationals ,  $\mathbb{Q}$ .

S = QQ

—A polynomial ring with lex ordering for  $y_1, y_2, \dots, y_m$ , lex ordering for the variable transformed phylogenetic tensor elements,  $q_1, q_2, \dots, q_p$ , which are indexed according to the ordering determined by the Kronecker product, and product ordering for  $y_1, y_2, \dots, y_m$ .

R = S[y-1..y-m, q-1..q-p, MonomialOrder => {Lex => m, Lex => p}]

—The generators of the ideal.

I = ideal(f1, f2, ..., fs)

—The generators of the ideal in the Gröbner basis, which have been transposed so they print down the page instead of across the page.

transpose gens gb I

For the three-taxon clock-like tree we have two time variables,  $m = 2$ , and three variable transformed phylogenetic tensor elements,  $p = 3$ . We will change the labelling of the transformed phylogenetic tensor elements from  $\{q_{011}, q_{101}, q_{110}\}$  to  $\{q_1, q_2, q_3\}$ . The *Macaulay2* code for the three-taxon clock-like tree is shown below.

S = QQ

R = S[y-1..y-2, q-1..q-3, MonomialOrder => {Lex => 2, Lex => 3}]

I = ideal(y-2^2 - q-1, y-1^2\*y-2^2 - q-2, y-1^2\*y-2^2 - q-3)

transpose gens gb I

The ideal in the Gröbner basis is found to be

$$I = \langle q_{101} - q_{110}, y_2^2 - q_{011}, y_1^2 q_{011} - q_{110} \rangle. \quad (6.12)$$

We will now check that this is indeed a reduced Gröbner basis for the ideal. It follows from Lemma 6.1.17 that to show that this is a Gröbner basis we need to show that the leading terms of all of the elements of  $I$  will be divisible by at least one of the leading monomials of the generators of the Gröbner basis,  $G$ . The ideal in the original basis can be expressed as

$$\begin{aligned} I &= \{h_1 (y_2^2 - q_{011}) + h_2 (y_1^2 y_2^2 - q_{101}) + h_3 (y_1^2 y_2^2 - q_{110}) : \\ &\quad h_1, h_2, h_3 \in \mathbb{F}[y_1, y_2, q_{011}, q_{101}, q_{110}]\} \\ &= \{(h_2 + h_3) y_1^2 y_2^2 + h_1 y_2^2 - h_1 q_{011} - h_2 q_{101} - h_3 q_{110} : \\ &\quad h_1, h_2, h_3 \in \mathbb{F}[y_1, y_2, q_{011}, q_{101}, q_{110}]\}. \end{aligned} \quad (6.13)$$

By restricting  $h_1, h_2, h_3$  to  $h_1, h_2, h_3 \in \{-1, 0, 1\}$  and taking the leading monomials we can find the set of all of the monomials that the leading terms of the elements of  $I$  will be divisible by. This set of monomials will be  $S = \{y_1^2 y_2^2, y_2^2, q_{101}\}$ . The leading terms of every element of  $I$  must be divisible by at least one of the monomials in  $S$ . For  $G$  to be a Gröbner basis, these three monomials in  $S$  must be divisible by at least one of the leading monomials of the generators of  $G$ . Clearly  $y_1^2 y_2^2$  will be divisible by  $y_2^2$ , while  $y_2^2$  will also be divisible by  $y_2^2$  and  $q_{101}$  will be divisible by  $q_{101}$ . Since the monomials of  $S$  will all be divisible by at least one of the leading monomials of the generators of  $G$  and every leading monomial of an element of  $I$  must be divisible by a monomial of  $S$  we can conclude that  $G$  is indeed a Gröbner basis for  $I$ .

We will now check that the Gröbner basis is a reduced Gröbner basis. The set of polynomial generators for the Gröbner basis is  $\{q_{101} - q_{110}, y_2^2 - q_{011}, y_1^2 q_{011} - q_{110}\}$ . We can see that  $LC(p) = 1$  for all of the polynomial generators, the first requirement for a reduced Gröbner basis. The first polynomial generator is  $p_1 = q_{101} - q_{110}$ . It follows that

$$\langle LT(G - \{p_1\}) \rangle = \langle y_2^2, y_1^2 q_{011} \rangle. \quad (6.14)$$

The two monomials of  $p_1$  are  $q_{101}$  and  $q_{110}$ .  $q_{101} \notin \{y_2^2, y_1^2 q_{011}\}$  and  $q_{110} \notin \{y_2^2, y_1^2 q_{011}\}$ . The second polynomial generator is  $p_2 = y_2^2 - q_{011}$ . It follows that

$$\langle LT(G - \{p_2\}) \rangle = \langle q_{101}, y_1^2 q_{011} \rangle. \quad (6.15)$$

The two monomials of  $p_2$  are  $y_2^2$  and  $q_{011}$ .  $y_2^2 \notin \{q_{101}, y_1^2 q_{011}\}$  and  $q_{011} \notin \{q_{101}, y_1^2 q_{011}\}$ . The third polynomial generator is  $p_3 = y_1^2 q_{011} - q_{110}$ . It follows that

$$\langle LT(G - \{p_3\}) \rangle = \langle q_{101}, y_2^2 \rangle. \quad (6.16)$$

The two monomials of  $p_3$  are  $y_1^2 q_{011}$  and  $q_{110}$ .  $y_1^2 q_{011} \notin \{q_{101}, y_2^2\}$  and  $q_{110} \notin \{q_{101}, y_2^2\}$ . We

can conclude that  $G$  is a reduced Gröbner basis for  $I$ .

Now solving this system of polynomial equations defined by the generators in the Gröbner basis gives

$$\left\{ q_{101} = q_{110}, \quad y_1 = \sqrt{\frac{q_{110}}{q_{011}}}, \quad y_2 = \sqrt{q_{011}} \right\}. \quad (6.17)$$

Recalling that  $0 < y_1 = e^{-\tau_1}, y_2 = e^{-\tau_2} \leq 1$ , the constraints on the non-constant elements of the transformed phylogenetic tensor for the non-clock-like tree are then

$$\{q_{101} = q_{110}, \quad q_{101} \leq q_{011}\}. \quad (6.18)$$

By using the same techniques, we will address the issue of identifiability for four-taxon trees and networks. We will see that this method will be much more efficient than finding the constraints manually when dealing with trees and networks with a large number of taxa or more convergence periods. We will then address the issue of distinguishability by finding the intersections of the constraints for pairs of trees or networks.

## 6.2 Four-Taxon Convergence-Divergence Networks

We now consider some four-taxon trees and networks as examples. There are two “structures” for four-taxon clock-like trees, defined by different sets of clusters. Below in Figure 6.2 are the two structures for four-taxon clock-like trees.

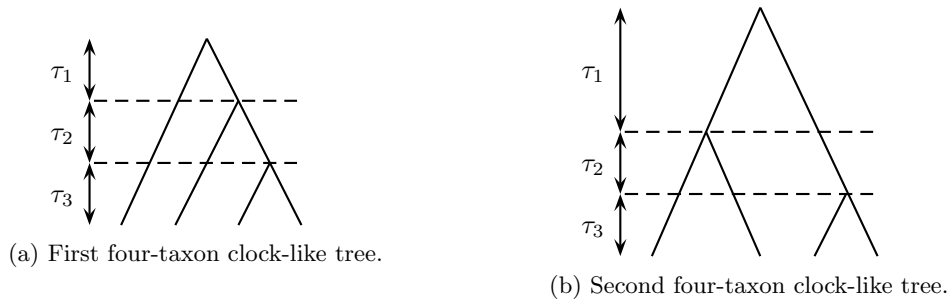


Figure 6.2: The two structures for four-taxon clock-like trees.

We will look at both of the four-taxon trees, as well as all of the convergence-divergence networks involving two leaves converging with each other and the non-clock-like tree. We give each tree or network an arbitrary name. For the first structure of the four-taxon clock-like tree there are four choices for pairs of converging leaves. When counting from the left, the inequivalent pairs of converging taxa are taxa 1 and 2, taxa 1 and 3, taxa 2 and 3 and taxa 3 and 4. Since the positioning of an edge below a node is arbitrary, the taxon pair 1 and 4 is equivalent to 1 and 3, while the taxon pair 2 and 4 is equivalent to 2 and 3. The first structure of the four-taxon

clock-like tree and the four convergence-divergence networks with pairs of converging leaves are shown in Figure 6.3 below.

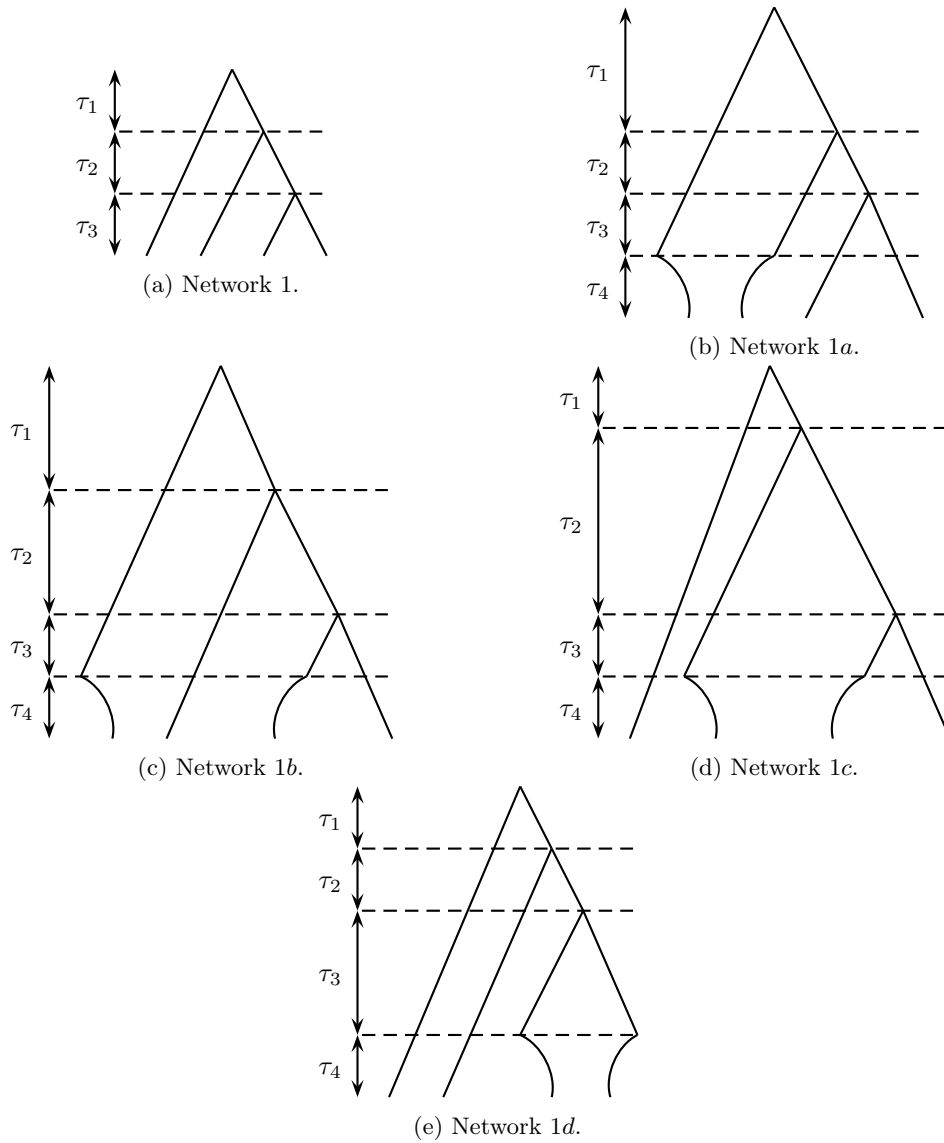


Figure 6.3: The first clock-like tree and the corresponding convergence-divergence networks.

There are three convergence-divergence networks with inequivalent pairs of leaves converging for the second structure of the four-taxon clock-like tree. The three inequivalent pairs of converging leaves are the two pairs of sister taxa convergence and the single non-sister taxa convergence pair. The four examples, including the second structure of the four-taxon clock-like tree, are shown below in Figure 6.4.

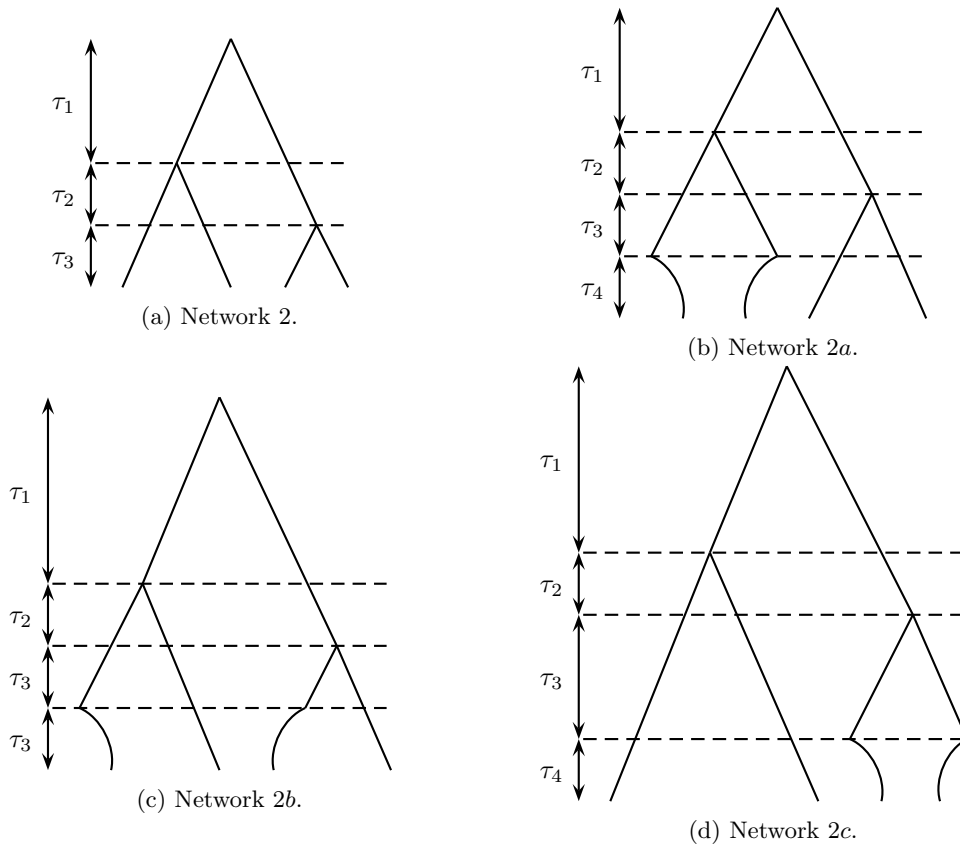


Figure 6.4: The second clock-like tree and the corresponding convergence-divergence networks.

Recall from Result 4.2.4.0.1 on page 79 that clock-like networks with convergence periods between sister taxa cannot be distinguished from the same networks with the convergence period replaced with a divergence period. From this result we can immediately rule that Network 1 and Network 1d are indistinguishable and disregard Network 1d. Similarly, Network 2a and Network 2c will both be indistinguishable from Network 2 and can be disregarded. This leaves six networks to examine: 1, 1a, 1b, 1c, 2 and 2b.

We will begin by stating the phylogenetic tensors in the Hadamard basis for each tree and network. Recall from Section 4.1 on page 69 that  $q_{0000} = 1$  and  $q_{0001} = q_{0010} = q_{0100} = q_{0111} = q_{1000} = q_{1011} = q_{1101} = q_{1110} = 0$ . Similarly to the two-taxon and three-taxon cases, for the binary symmetric model the transformed phylogenetic tensor elements for each tree and network that we will examine will satisfy the constraints,

$$\left\{ \begin{array}{ll} q_{0000} &= 1, \\ q_{0001} = q_{0010} = q_{0100} = q_{0111} = q_{1000} = q_{1011} = q_{1101} = q_{1110} &= 0, \\ 0 < q_{0011}, q_{0101}, q_{0110}, q_{1001}, q_{1010}, q_{1100}, q_{1111} &\leq 1. \end{array} \right. \quad (6.19)$$

Consequently, we will not consider the transformed phylogenetic tensor elements,  $q_{0000}$ ,  $q_{0001}$ ,  $q_{0010}$ ,  $q_{0100}$ ,  $q_{0111}$ ,  $q_{1000}$ ,  $q_{1011}$ ,  $q_{1101}$  and  $q_{1110}$  since they are uninformative.

We will then find the remaining constraints on the transformed phylogenetic tensor elements for each tree and network from their Gröbner bases before comparing these constraints between trees and networks. We will only write down the Gröbner bases for some of the trees and networks, since for the non-clock-like tree and some of the convergence-divergence networks there are many generators with many monomials in the Gröbner basis. The Gröbner bases that are not shown in Chapter 6 will be shown in Appendices E,F,G and H.

### 6.2.1 Network 1

The first tree or network that we will examine is Network 1, shown in Figure 6.5 below.

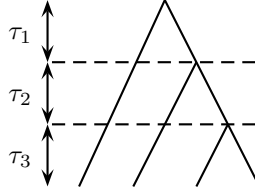


Figure 6.5: Network 1.

The variable transformed phylogenetic tensor elements are

$$\begin{bmatrix} q_{0011} \\ q_{0101} \\ q_{0110} \\ q_{1001} \\ q_{1010} \\ q_{1100} \\ q_{1111} \end{bmatrix} = \begin{bmatrix} e^{-2\tau_3} \\ e^{-2(\tau_2+\tau_3)} \\ e^{-2(\tau_2+\tau_3)} \\ e^{-2(\tau_1+\tau_2+\tau_3)} \\ e^{-2(\tau_1+\tau_2+\tau_3)} \\ e^{-2(\tau_1+\tau_2+\tau_3)} \\ e^{-2(\tau_1+\tau_2+2\tau_3)} \end{bmatrix} = \begin{bmatrix} y_3^2 \\ y_2^2 y_3^2 \\ y_2^2 y_3^2 \\ y_1^2 y_2^2 y_3^2 \\ y_1^2 y_2^2 y_3^2 \\ y_1^2 y_2^2 y_3^2 \\ y_1^2 y_2^2 y_3^4 \end{bmatrix}. \quad (6.20)$$

We form the ideal,

$$I = \langle y_3^2 - q_{0011}, y_2^2 y_3^2 - q_{0101}, y_2^2 y_3^2 - q_{0110}, y_1^2 y_2^2 y_3^2 - q_{1001}, y_1^2 y_2^2 y_3^2 - q_{1010}, y_1^2 y_2^2 y_3^2 - q_{1100}, y_1^2 y_2^2 y_3^4 - q_{1111} \rangle. \quad (6.21)$$

Using the *Macaulay2* code from Section 6.1 on page 125, the Gröbner basis of the ideal for Network 1 is

$$I = \langle q_{1010} - q_{1100}, q_{1001} - q_{1100}, q_{0101} - q_{0110}, q_{0011} q_{1100} - q_{1111}, y_3^2 - q_{0011}, y_2^2 q_{1111} - q_{0110} q_{1100}, y_2^2 q_{0011} - q_{0110}, y_1^2 q_{0110} - q_{1100} \rangle. \quad (6.22)$$



After finding the Gröbner basis of the ideal, solving for the time parameters,

$$\begin{cases} y_1 &= \sqrt{\frac{q_{1001}}{q_{0101}}}, \\ y_2 &= \sqrt{\frac{q_{0101}}{q_{0011}}}, \\ y_3 &= \sqrt{q_{0011}}. \end{cases} \quad (6.23)$$

Demanding  $0 < y_1, y_2, y_3 \leq 1$  from  $y_i = e^{-\tau_i}$ , where  $\tau_i \geq 0$  is a time parameter, the constraints on Network 1 are then

$$\begin{cases} q_{0101} &= q_{0110}, \\ q_{1001} &= q_{1010}, \\ q_{1001} &= q_{1100}, \\ q_{0011}q_{1001} &= q_{1111}, \\ q_{0101} &\leq q_{0011}, \\ q_{1001} &\leq q_{0101}. \end{cases} \quad (6.24)$$

Recall from Definition 5.5 on page 105 that the distance between any two leaves on a network is called the pairwise distance. The pairwise distance between two leaves,  $a$  and  $b$ , is defined as

$$d(a, b) = -\ln(q_A), \quad (6.25)$$

where  $A = i_1 i_2 \dots i_n$ ,  $a, b \in \{1, 2, \dots, n\}$ ,  $a \neq b$ ,  $i_a = i_b = 1$  and  $i_k = 0$  for all  $k \in \{1, 2, \dots, n\} \setminus \{a, b\}$ .

The first equality,  $q_{0101} = q_{0110}$ , implies that the pairwise distance between taxa 2 and 4 must be equal to the pairwise distance between taxa 2 and 3,  $d(2, 4) = d(2, 3)$ . The second equality,  $q_{1001} = q_{1010}$ , implies that the pairwise distance between taxa 1 and 4 must be equal to the pairwise distance between taxa 1 and 3,  $d(1, 4) = d(1, 3)$ . These pairs of equal distances are obvious since taxa 3 and 4 are sister taxa. The third equality,  $q_{1001} = q_{1100}$ , implies that the pairwise distance between taxa 1 and taxa 4 must be equal to the pairwise distance between taxa 1 and 2,  $d(1, 4) = d(1, 2)$ . The pairwise distances are the sums of the edge lengths connecting the two leaves. We can see that these two pairwise distances must be equal by looking at Figure 6.5.

We will now look at the fourth equality,  $q_{0011}q_{1001} = q_{1111}$ . We have defined pairwise distances in terms of the transformed phylogenetic tensor elements with two elements being one and the rest being zero. It is not immediately clear how to interpret this equality since we have not defined any of the pairwise distances in terms of  $q_{1111}$ . Focussing on the left hand side of the equality,

$$q_{0011}q_{1001} = e^{-d(3,4)}e^{-d(1,4)} = e^{-(d(3,4)+d(1,4))} = e^{-2(\tau_1+\tau_2+2\tau_3)} = q_{1111}. \quad (6.26)$$

We will now define a *total network distance* in a similar way to how we defined tree distances and pairwise distances.

**Definition 6.2.1 (Total Network Distance).** *The **total network distance**,  $d(1, 2, 3, 4)$ , on a four-taxon convergence network is expressed as*

$$d(1, 2, 3, 4) = -\ln(q_{1111}).$$

Notice that on Network 1, the total network distance is  $d(1, 2, 3, 4) = d(3, 4) + d(1, 4) = d(3, 4) + d(1, 2)$ , which is the sum of the pairwise distances either side of the cluster, [12|34].

The first inequality,  $q_{0101} \leq q_{0011}$ , implies that the pairwise distance between taxa 2 and 4 must be greater than the pairwise distance between taxa 3 and 4, which can again be seen by looking at Figure 6.5. The second inequality,  $q_{1001} \leq q_{0101}$ , implies that the pairwise distance between taxa 1 and 4 must be greater than the pairwise distance between taxa 2 and 4, which can also be seen by looking at Figure 6.5. In summary,

$$\begin{cases} d(3, 4) \leq d(2, 4) = d(2, 3) \leq d(1, 4) = d(1, 3) & = d(1, 2), \\ d(3, 4) + d(1, 2) & = d(1, 2, 3, 4). \end{cases} \quad (6.27)$$

Semple and Steel [2003] defined the four-point condition in terms of dissimilarity maps on a tree. We will define it in terms of the sums of pairwise distance.

**Theorem 6.2.2 (Four-Point Condition).** *Given three different pairwise distances,  $d_A$ ,  $d_B$  and  $d_C$ , where  $d_A, d_B, d_C \in \{d(3, 4) + d(1, 2), d(2, 4) + d(1, 3), d(2, 3) + d(1, 4)\}$ ,*

$$d_A \leq d_B = d_C.$$

*The four-point condition is satisfied if we have a phylogenetic tree.*

*Proof.* Proof can be found in Semple and Steel [2003]. □

Notice that on Network 1,

$$\begin{aligned} d(3, 4) + d(1, 2) &= 2(\tau_1 + \tau_2 + 2\tau_3) \\ &\leq d(2, 4) + d(1, 3) = d(2, 3) + d(1, 4) \\ &= 2(\tau_1 + 2(\tau_2 + \tau_3)) \\ &= 2(\tau_1 + \tau_2 + 2\tau_3) + 2\tau_2 \\ &= d(3, 4) + d(1, 2) + 2\tau_2. \end{aligned} \quad (6.28)$$

We can conclude that Network 1 satisfies the four-point condition, which is to be expected since it is a tree.

We can use the four-point condition as a relatively simple check for distinguishability. If a given network does not satisfy the four-point condition then it will be distinguishable from

trees. The reverse is not generally true, however. A tree or network can be distinguishable from another tree or network despite both satisfying the four-point condition.

### 6.2.2 Network 1a

The next tree or network that we will examine is Network 1a, shown in Figure 6.6 below.

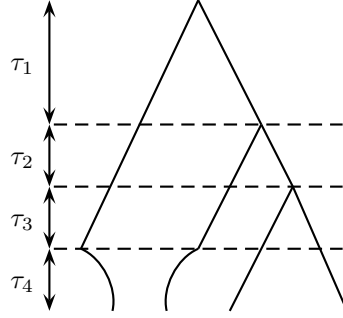


Figure 6.6: Network 1a.

The variable transformed phylogenetic tensor elements are

$$\begin{aligned}
 \begin{bmatrix} q_{0011} \\ q_{0101} \\ q_{0110} \\ q_{1001} \\ q_{1010} \\ q_{1100} \\ q_{1111} \end{bmatrix} &= \begin{bmatrix} e^{-2(\tau_3+\tau_4)} \\ e^{-2(\tau_2+\tau_3+\tau_4)} \\ e^{-2(\tau_2+\tau_3+\tau_4)} \\ e^{-2(\tau_1+\tau_2+\tau_3+\tau_4)} \\ e^{-2(\tau_1+\tau_2+\tau_3+\tau_4)} \\ 1 - e^{-\tau_4} (1 - e^{-2(\tau_1+\tau_2+\tau_3)}) \\ e^{-2(\tau_1+\tau_2+\tau_3+\tau_4)} (1 - e^{-\tau_4} (1 - e^{-2(\tau_1+\tau_2+\tau_3)})) \end{bmatrix} \\
 &= \begin{bmatrix} y_3^2 y_4^2 \\ y_2^2 y_3^2 y_4^2 \\ y_2^2 y_3^2 y_4^2 \\ y_1^2 y_2^2 y_3^2 y_4^2 \\ y_1^2 y_2^2 y_3^2 y_4^2 \\ 1 - y_4 (1 - y_1^2 y_2^2 y_3^2) \\ y_3^2 y_4^2 (1 - y_4 (1 - y_1^2 y_2^2 y_3^2)) \end{bmatrix}.
 \end{aligned} \tag{6.29}$$

We form the ideal,

$$\begin{aligned}
 I = \langle & y_3^2 y_4^2 - q_{0011}, y_2^2 y_3^2 y_4^2 - q_{0101}, y_2^2 y_3^2 y_4^2 - q_{0110}, y_1^2 y_2^2 y_3^2 y_4^2 - q_{1001}, y_1^2 y_2^2 y_3^2 y_4^2 - q_{1010}, \\
 & 1 - y_4 (1 - y_1^2 y_2^2 y_3^2) - q_{1100}, y_3^2 y_4^2 (1 - y_4 (1 - y_1^2 y_2^2 y_3^2)) - q_{1111} \rangle.
 \end{aligned} \tag{6.30}$$

Using the *Macaulay2* code from Section 6.1 on page 125, the Gröbner basis of the ideal for

Network 1a is

$$\begin{aligned}
I = \langle & q_{1001} - q_{1010}, q_{0101} - q_{0110}, q_{0011}q_{1100} - q_{1111}, y_4^2 + y_4q_{1100} - y_4 - q_{1010}, \\
& y_3^2q_{1010} + y_4q_{0011} - y_4q_{1111} - q_{0011}q_{1010} - q_{0011} - q_{1100}q_{1111} + 2q_{1111}, \\
& y_3^2y_4q_{1100} - y_3^2y_4 - y_3^2q_{1010} + q_{0011}, y_3^2y_4q_{1010} - y_4q_{0011} + q_{0011} - q_{1111}, \\
& y_3^2y_4q_{0011} - y_3^2y_4q_{1111} + y_3^2q_{0011}q_{1010} - q_{0011}^2, y_2^2q_{1111} - q_{0110}q_{1100}, y_2^2q_{0011} - q_{0110}, \\
& y_1^2q_{0110} - q_{1010}, y_1^2y_2^2y_3^2q_{1010} - y_4q_{1100} + y_4 - q_{1010} - q_{1100}^2 + 2q_{1100} - 1, \\
& y_1^2y_2^2y_3^2y_4 - y_4 - q_{1100} + 1 \rangle.
\end{aligned} \tag{6.31}$$

After finding the Gröbner basis of the ideal, solving for the time parameters,

$$\begin{cases}
y_1 &= \sqrt{\frac{q_{1001}}{q_{0101}}}, \\
y_2 &= \sqrt{\frac{q_{0101}}{q_{0011}}}, \\
y_3 &= \frac{\sqrt{q_{0011}} \left( -(1 - q_{1100}) + \sqrt{(1 - q_{1100})^2 + 4q_{1001}} \right)}{2q_{1001}}, \\
y_4 &= \frac{1}{2} \left( (1 - q_{1100}) + \sqrt{(1 - q_{1100})^2 + 4q_{1001}} \right).
\end{cases} \tag{6.32}$$

We demand  $0 < y_1, y_2, y_3, y_4 \leq 1$  from  $y_i = e^{-\tau_i}$ . Demanding  $y_1 \leq 1$  implies

$$q_{1001} \leq q_{0101}. \tag{6.33}$$

Demanding  $y_2 \leq 1$  implies

$$q_{0101} \leq q_{0011}. \tag{6.34}$$

Now suppose we have some expression similar to the expression for  $y_4$  in the form

$$y_i = \frac{1}{2} \left( (1 - q_a) + \sqrt{(1 - q_a)^2 + 4q_b} \right), \tag{6.35}$$

where  $y_i$  is a time parameter and  $q_a$  and  $q_b$  are transformed phylogenetic tensor elements. Demanding  $y_i \leq 1$ ,

$$\begin{aligned}
\frac{1}{2} \left( (1 - q_a) + \sqrt{(1 - q_a)^2 + 4q_b} \right) &\leq 1 \\
\Leftrightarrow (1 - q_a) + \sqrt{(1 - q_a)^2 + 4q_b} &\leq 2 \\
\Leftrightarrow \sqrt{(1 - q_a)^2 + 4q_b} &\leq 1 + q_a.
\end{aligned} \tag{6.36}$$

Since  $q_a, q_b > 0$ ,

$$\begin{aligned}
 &\Leftrightarrow (1 - q_a)^2 + 4q_b \leq (1 + q_a)^2 \\
 &\Leftrightarrow 1 + q_a^2 - 2q_a + 4q_b \leq 1 + q_a^2 + 2q_a \\
 &\Leftrightarrow 4q_b \leq 4q_a \\
 &\Leftrightarrow q_b \leq q_a.
 \end{aligned} \tag{6.37}$$

From (6.37) we can conclude that

$$q_{1001} \leq q_{1100}. \tag{6.38}$$

Now suppose we have an expression similar to the expression for  $y_3$  in the form

$$y_j = \frac{\sqrt{q_e} \left( -(1 - q_c) + \sqrt{(1 - q_c)^2 + 4q_d} \right)}{2q_d}. \tag{6.39}$$

Now, demanding  $y_b \leq 1$ ,

$$\frac{\sqrt{q_e} \left( -(1 - q_c) + \sqrt{(1 - q_c)^2 + 4q_d} \right)}{2q_d} \leq 1. \tag{6.40}$$

Since  $q_d > 0$ ,

$$\sqrt{q_e} \left( -(1 - q_c) + \sqrt{(1 - q_c)^2 + 4q_d} \right) \leq 2q_d. \tag{6.41}$$

Since  $q_e > 0$ ,

$$\begin{aligned}
 -(1 - q_c) + \sqrt{(1 - q_c)^2 + 4q_d} &\leq \frac{2q_d}{\sqrt{q_e}} \\
 \Leftrightarrow \sqrt{(1 - q_c)^2 + 4q_d} &\leq \frac{2q_d}{\sqrt{q_e}} + (1 - q_c).
 \end{aligned} \tag{6.42}$$

Since  $q_c \leq 1$ ,

$$\begin{aligned}
(1 - q_c)^2 + 4q_d &\leq \frac{4q_d^2}{q_e} + (1 - q_c)^2 + \frac{4q_d(1 - q_c)}{\sqrt{q_e}} \\
&\Leftrightarrow 4q_d \leq \frac{4q_d^2}{q_e} + \frac{4q_d(1 - q_c)}{\sqrt{q_e}} \\
&\Leftrightarrow 1 \leq \frac{q_d}{q_e} + \frac{(1 - q_c)}{\sqrt{q_e}} \\
&\Leftrightarrow q_e \leq q_d + \sqrt{q_e}(1 - q_c) \\
&\Leftrightarrow (q_e - q_d) \leq \sqrt{q_e}(1 - q_c).
\end{aligned} \tag{6.43}$$

If  $q_d \leq q_e$  then

$$\Leftrightarrow (q_e - q_d)^2 \leq q_e(1 - q_c)^2. \tag{6.44}$$

Since from (6.33),  $q_{1001} \leq q_{0101}$ , and from (6.34),  $q_{0101} \leq q_{0011}$ , then

$$q_{1001} \leq q_{0011}. \tag{6.45}$$

From (6.43) and (6.44) we can conclude that

$$(q_{0011} - q_{1001})^2 \leq q_{0011}(1 - q_{1100})^2. \tag{6.46}$$

The constraints on Network 1a are then

$$\left\{ \begin{array}{ll} q_{0101} &= q_{0110}, \\ q_{1001} &= q_{1010}, \\ q_{0011}q_{1100} &= q_{1111}, \\ q_{0101} &\leq q_{0011}, \\ q_{1001} &\leq q_{0101}, \\ q_{1001} &\leq q_{1100}, \\ (q_{0011} - q_{1001})^2 &\leq q_{0011}(1 - q_{1100})^2. \end{array} \right. \tag{6.47}$$

The constraints on the distances for Network 1a that are equivalent to the constraints on the distances for Network 1 are

$$\left\{ \begin{array}{ll} d(3, 4) \leq d(2, 4) = d(2, 3) \leq d(1, 4) &= d(1, 3), \\ d(3, 4) + d(1, 2) &= d(1, 2, 3, 4). \end{array} \right. \tag{6.48}$$

Network 1a also has the distance constraint,  $d(1, 2) \leq d(1, 4)$ , while on Network 1,  $d(1, 2) = d(1, 4)$ . We will see later how the last inequality constraint compares to the constraints for the

other trees and networks.

From  $d(1, 3) = d(1, 4)$  and  $d(2, 4) = d(2, 3)$  we can conclude that

$$d(2, 4) + d(1, 3) = d(2, 3) + d(1, 4). \quad (6.49)$$

From  $d(3, 4) \leq d(2, 3)$  and  $d(1, 2) \leq d(1, 4)$  we can conclude that

$$d(3, 4) + d(1, 2) \leq d(2, 3) + d(1, 4). \quad (6.50)$$

Despite being a convergence-divergence network and not a tree, we can conclude that Network 1a must satisfy the four-point condition,

$$d(3, 4) + d(1, 2) \leq d(2, 4) + d(1, 3) = d(2, 3) + d(1, 4). \quad (6.51)$$

### 6.2.3 Network 1b

The next tree or network that we will examine is Network 1b, shown in Figure 6.7 below.

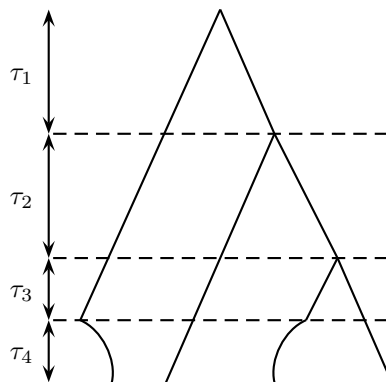


Figure 6.7: Network 1b.

The variable transformed phylogenetic tensor elements are

$$\begin{aligned}
 \begin{bmatrix} q_{0011} \\ q_{0101} \\ q_{0110} \\ q_{1001} \\ q_{1010} \\ q_{1100} \\ q_{1111} \end{bmatrix} &= \begin{bmatrix} e^{-2(\tau_3+\tau_4)} \\ e^{-2(\tau_2+\tau_3+\tau_4)} \\ e^{-2(\tau_2+\tau_3+\tau_4)} \\ 1 - e^{-\tau_4} (1 - e^{-2(\tau_1+\tau_2+\tau_3)}) \\ e^{-2(\tau_1+\tau_2+\tau_3+\tau_4)} \\ e^{-2(\tau_1+\tau_2+\tau_3+\tau_4)} \\ e^{-2(\tau_2+\tau_3+\tau_4)} (1 - e^{-\tau_4} (1 - e^{-2(\tau_1+\tau_3)})) \end{bmatrix} \\
 &= \begin{bmatrix} y_3^2 y_4^2 \\ y_2^2 y_3^2 y_4^2 \\ y_2^2 y_3^2 y_4^2 \\ 1 - y_4 (1 - y_1^2 y_2^2 y_3^2) \\ y_1^2 y_2^2 y_3^2 y_4^2 \\ y_1^2 y_2^2 y_3^2 y_4^2 \\ y_2^2 y_3^2 y_4^2 (1 - y_4 (1 - y_1^2 y_3^2)) \end{bmatrix}.
 \end{aligned} \tag{6.52}$$

We form the ideal,

$$\begin{aligned}
 I = \langle & y_3^2 y_4^2 - q_{0011}, y_2^2 y_3^2 y_4^2 - q_{0101}, y_2^2 y_3^2 y_4^2 - q_{0110}, 1 - y_4 (1 - y_1^2 y_2^2 y_3^2) - q_{1001}, y_1^2 y_2^2 y_3^2 y_4^2 - q_{1010}, \\
 & y_1^2 y_2^2 y_3^2 y_4^2 - q_{1100}, y_2^2 y_3^2 y_4^2 (1 - y_4 (1 - y_1^2 y_3^2)) - q_{1111} \rangle.
 \end{aligned} \tag{6.53}$$

The Gröbner basis of the ideal is given in Appendix E, it has 24 generators. After finding the Gröbner basis of the ideal, solving for the time parameters,

$$\begin{cases} y_1 &= \sqrt{\frac{q_{1010}}{q_{0101}}}, \\ y_2 &= \sqrt{\frac{q_{0101}}{q_{0011}}}, \\ y_3 &= \frac{\sqrt{q_{0011}} \left( -(1 - q_{1001}) + \sqrt{(1 - q_{1001})^2 + 4q_{1010}} \right)}{2q_{1010}}, \\ y_4 &= \frac{1}{2} \left( (1 - q_{1001}) + \sqrt{(1 - q_{1001})^2 + 4q_{1010}} \right). \end{cases} \tag{6.54}$$

Demanding  $y_1, y_2 \leq 1$  and from (6.37), (6.43) and (6.44), the constraints on Network 1b are



then

$$\left\{ \begin{array}{l} q_{0101} = q_{0110}, \\ q_{1010} = q_{1100}, \\ q_{0011}^2 q_{1010} - q_{0011} q_{0101} q_{1001}^2 + q_{0011} q_{0101} q_{1001} - 2q_{0011} q_{0101} q_{1010} \\ + q_{0011} q_{1001} q_{1111} - q_{0011} q_{1111} - q_{0101}^2 q_{1001} + q_{0101}^2 q_{1010} \\ + q_{0101} q_{1001} q_{1111} + q_{0101} q_{1111} - q_{1111}^2 = 0, \\ q_{0101} \leq q_{0011}, \\ q_{1010} \leq q_{0101}, \\ q_{1010} \leq q_{1001}, \\ (q_{0011} - q_{1010})^2 \leq q_{0011} (1 - q_{1001})^2. \end{array} \right. \quad (6.55)$$

The constraints on the distances for Network 1b that are equivalent to the constraints on the distances for Network 1 are

$$d(3, 4) \leq d(2, 4) = d(2, 3) \leq d(1, 3) = d(1, 2). \quad (6.56)$$

Network 1b also has the distance constraint,  $d(1, 4) \leq d(1, 3)$ , while on Network 1,  $d(1, 4) = d(1, 3)$ . While on Network 1 the total network distance satisfies  $d(1, 2, 3, 4) = d(3, 4) + d(1, 2)$ , on Network 1a,  $d(1, 2, 3, 4) \neq d(3, 4) + d(1, 2)$ .

From  $d(3, 4) \leq d(2, 4)$  and  $d(1, 2) = d(1, 3)$  we can conclude that

$$d(3, 4) + d(1, 2) \leq d(2, 4) + d(1, 3). \quad (6.57)$$

From  $d(2, 4) = d(2, 3)$  and  $d(1, 4) \leq d(1, 3)$  we can conclude that

$$d(2, 3) + d(1, 4) \leq d(2, 4) + d(1, 3). \quad (6.58)$$

We can conclude that the sum of pairwise distances,  $d(2, 4) + d(1, 3)$ , must be greater than or equal to each of the other two sums of pairwise distances. For the four-point condition to be satisfied the two greatest sums of pairwise distances must be equal. We can conclude that Network 1b does not generally satisfy the four-point condition.

#### 6.2.4 Network 1c

The next tree or network that we will examine is Network 1c, shown in Figure 6.8 below.

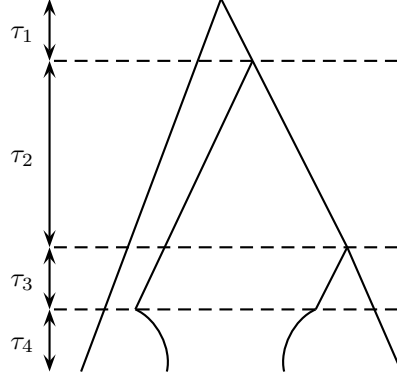


Figure 6.8: Network 1c.

The variable transformed phylogenetic tensor elements are

$$\begin{aligned}
 \begin{bmatrix} q_{0011} \\ q_{0101} \\ q_{0110} \\ q_{1001} \\ q_{1010} \\ q_{1100} \\ q_{1111} \end{bmatrix} &= \begin{bmatrix} e^{-2(\tau_3+\tau_4)} \\ 1 - e^{-\tau_4} (1 - e^{-2(\tau_2+\tau_3)}) \\ e^{-2(\tau_2+\tau_3+\tau_4)} \\ e^{-2(\tau_1+\tau_2+\tau_3+\tau_4)} \\ e^{-2(\tau_1+\tau_2+\tau_3+\tau_4)} \\ e^{-2(\tau_1+\tau_2+\tau_3+\tau_4)} \\ e^{-2(\tau_1+\tau_2+\tau_3+\tau_4)} (1 - e^{-\tau_4} (1 - e^{-2\tau_3})) \end{bmatrix} \\
 &= \begin{bmatrix} y_3^2 y_4^2 \\ 1 - y_4 (1 - y_2^2 y_3^2) \\ y_2^2 y_3^2 y_4^2 \\ y_1^2 y_2^2 y_3^2 y_4^2 \\ y_1^2 y_2^2 y_3^2 y_4^2 \\ y_1^2 y_2^2 y_3^2 y_4^2 \\ y_1^2 y_2^2 y_3^2 y_4^2 (1 - y_4 (1 - y_3^2)) \end{bmatrix}.
 \end{aligned} \tag{6.59}$$

We form the ideal,

$$\begin{aligned}
 I = \langle y_3^2 y_4^2 - q_{0011}, 1 - y_4 (1 - y_2^2 y_3^2) - q_{0101}, y_2^2 y_3^2 y_4^2 - q_{0110}, y_1^2 y_2^2 y_3^2 y_4^2 - q_{1001}, y_1^2 y_2^2 y_3^2 y_4^2 - q_{1010}, \\
 y_1^2 y_2^2 y_3^2 y_4^2 - q_{1100}, y_1^2 y_2^2 y_3^2 y_4^2 (1 - y_4 (1 - y_3^2)) - q_{1111} \rangle.
 \end{aligned} \tag{6.60}$$

The Gröbner basis of the ideal is given in Appendix F, it has 30 generators. After finding the

Gröbner basis of the ideal, solving for the time parameters,

$$\begin{cases} y_1 &= \sqrt{\frac{q_{1001}}{q_{0110}}}, \\ y_2 &= \sqrt{\frac{q_{0110}}{q_{0011}}}, \\ y_3 &= \frac{\sqrt{q_{0011}} \left( -(1-q_{0101}) + \sqrt{(1-q_{0101})^2 + 4q_{0110}} \right)}{2q_{0110}}, \\ y_4 &= \frac{1}{2} \left( (1-q_{0101}) + \sqrt{(1-q_{0101})^2 + 4q_{0110}} \right). \end{cases} \quad (6.61)$$

Demanding  $y_1, y_2 \leq 1$  and from (6.37), (6.43) and (6.44), the constraints on Network 1c are then

$$\left\{ \begin{array}{ll} q_{1001} &= q_{1010}, \\ q_{1001} &= q_{1100}, \\ q_{0011}^2 q_{1001}^2 - q_{0011} q_{0101}^2 q_{1001}^2 + q_{0011} q_{0101} q_{1001}^2 + q_{0011} q_{0101} q_{1001} q_{1111} \\ &\quad - 2q_{0011} q_{0110} q_{1001}^2 - q_{0011} q_{1001} q_{1111} - q_{0101} q_{0110} q_{1001}^2 \\ &\quad + q_{0101} q_{0110} q_{1001} q_{1111} + q_{0110}^2 q_{1001}^2 + q_{0110} q_{1001} q_{1111} - q_{0110} q_{1111}^2 = 0, \\ q_{0110} &\leq q_{0011}, \\ q_{0110} &\leq q_{0101}, \\ q_{1001} &\leq q_{0110}, \\ (q_{0011} - q_{0110})^2 &\leq q_{0011} (1 - q_{0101})^2. \end{array} \right. \quad (6.62)$$

The constraints on the distances for Network 1c that are equivalent to the constraints on the distances for Network 1 are

$$d(3, 4) \leq d(2, 3) \leq d(1, 4) = d(1, 3) = d(1, 2). \quad (6.63)$$

Network 1c also has the distance constraint,  $d(2, 4) \leq d(2, 3)$ , while on Network 1,  $d(2, 4) = d(2, 3)$ . While on Network 1 the total network distance satisfies  $d(1, 2, 3, 4) = d(3, 4) + d(1, 2)$ , on Network 1c,  $d(1, 2, 3, 4) \neq d(3, 4) + d(1, 2)$ . We will see later how the quintic equality constraint with 11 terms compares to the constraints for the other trees and networks.

From  $d(3, 4) \leq d(2, 3)$  and  $d(1, 2) = d(1, 4)$  we can conclude that

$$d(3, 4) + d(1, 2) \leq d(2, 3) + d(1, 4). \quad (6.64)$$

From  $d(2, 4) \leq d(2, 3)$  and  $d(1, 3) = d(1, 4)$  we can conclude that

$$d(2, 4) + d(1, 3) \leq d(2, 3) + d(1, 4). \quad (6.65)$$

We can conclude that the sum of pairwise distances,  $d(2,3) + d(1,4)$ , must be greater than or equal to each of the other two sums of pairwise distances. For the four-point condition to be satisfied the two greatest sums of pairwise distances must be equal. We can conclude that Network 1c does not generally satisfy the four-point condition.

### 6.2.5 Network 2

The next tree or network that we will examine is Network 2, shown in Figure 6.9 below.

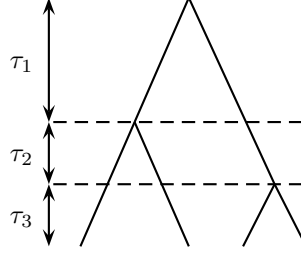


Figure 6.9: Network 2.

The variable transformed phylogenetic tensor elements are

$$\begin{bmatrix} q_{0011} \\ q_{0101} \\ q_{0110} \\ q_{1001} \\ q_{1010} \\ q_{1100} \\ q_{1111} \end{bmatrix} = \begin{bmatrix} e^{-2\tau_3} \\ e^{-2(\tau_1+\tau_2+\tau_3)} \\ e^{-2(\tau_1+\tau_2+\tau_3)} \\ e^{-2(\tau_1+\tau_2+\tau_3)} \\ e^{-2(\tau_1+\tau_2+\tau_3)} \\ e^{-2(\tau_2+\tau_3)} \\ e^{-2(\tau_2+2\tau_3)} \end{bmatrix} = \begin{bmatrix} y_3^2 \\ y_1^2 y_2^2 y_3^2 \\ y_1^2 y_2^2 y_3^2 \\ y_1^2 y_2^2 y_3^2 \\ y_1^2 y_2^2 y_3^2 \\ y_2^2 y_3^2 \\ y_2^2 y_3^4 \end{bmatrix}. \quad (6.66)$$

We form the ideal,

$$I = \langle y_3^2 - q_{0011}, y_1^2 y_2^2 y_3^2 - q_{0101}, y_1^2 y_2^2 y_3^2 - q_{0110}, y_1^2 y_2^2 y_3^2 - q_{1001}, y_1^2 y_2^2 y_3^2 - q_{1010}, y_2^2 y_3^2 - q_{1100}, y_2^2 y_3^4 - q_{1111} \rangle. \quad (6.67)$$

Using the *Macaulay2* code from Section 6.1 on page 125, the Gröbner basis of the ideal for Network 2 is

$$I = \langle q_{1001} - q_{1010}, q_{0110} - q_{1010}, q_{0101} - q_{1010}, q_{0011} q_{1100} - q_{1111}, y_3^2 - q_{0011}, y_2^2 q_{1111} - q_{1100}^2, y_2^2 q_{0011} - q_{1100}, y_1^2 q_{1111} - q_{0011} q_{1010}, y_1^2 q_{1100} - q_{1010} \rangle. \quad (6.68)$$

After finding the Gröbner basis of the ideal, solving for the time parameters,

$$\begin{cases} y_1 &= \sqrt{\frac{q_{0101}}{q_{1100}}}, \\ y_2 &= \sqrt{\frac{q_{1100}}{q_{0011}}}, \\ y_3 &= \sqrt{q_{0011}}. \end{cases} \quad (6.69)$$

Demanding  $y_1, y_2 \leq 1$ , the constraints on Network 2 are then

$$\begin{cases} q_{0101} &= q_{0110}, \\ q_{0101} &= q_{1001}, \\ q_{0101} &= q_{1010}, \\ q_{0011}q_{1100} &= q_{1111}, \\ q_{0101} &\leq q_{1100}, \\ q_{1100} &\leq q_{0011}. \end{cases} \quad (6.70)$$

The constraints on the distances for Network 2 that are equivalent to the constraints on the distances for Network 1 are

$$\begin{cases} d(3, 4) \leq d(2, 4) &= d(2, 3), \\ d(3, 4) &\leq d(1, 2), \\ d(1, 4) &= d(1, 3), \\ d(3, 4) + d(1, 2) &= d(1, 2, 3, 4). \end{cases} \quad (6.71)$$

Network 2 also has the distance constraints,  $d(1, 2) \leq d(2, 4) = d(1, 4)$ , while on Network 1,  $d(2, 4) \leq d(1, 4) = d(1, 2)$ .

Notice that on Network 2,

$$\begin{aligned} d(3, 4) + d(1, 2) &= 2(\tau_2 + 2\tau_3) \\ &\leq d(2, 4) + d(1, 3) = d(2, 3) + d(1, 4) \\ &= 4(\tau_1 + \tau_2 + \tau_3) \\ &= 2(\tau_2 + 2\tau_3) + 2(2\tau_1 + \tau_2) \\ &= d(3, 4) + d(1, 2) + 2(2\tau_1 + \tau_2). \end{aligned} \quad (6.72)$$

We can conclude that Network 2 satisfies the four-point condition, which is to be expected since it is a tree.

### 6.2.6 Network 2b

The next tree or network that we will examine is Network 2b, shown in Figure 6.10 below.

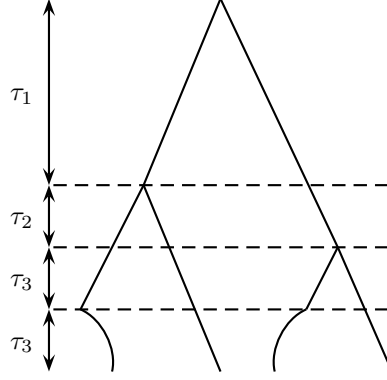


Figure 6.10: Network 2b.

The variable transformed phylogenetic tensor elements are

$$\begin{aligned}
 \begin{bmatrix} q_{0011} \\ q_{0101} \\ q_{0110} \\ q_{1001} \\ q_{1010} \\ q_{1100} \\ q_{1111} \end{bmatrix} &= \begin{bmatrix} e^{-2(\tau_3+\tau_4)} \\ e^{-2(\tau_1+\tau_2+\tau_3+\tau_4)} \\ e^{-2(\tau_1+\tau_2+\tau_3+\tau_4)} \\ e^{-2(\tau_1+\tau_2+\tau_3+\tau_4)} \\ 1 - e^{-\tau_4} (1 - e^{-2(\tau_1+\tau_2+\tau_3)}) \\ e^{-2(\tau_2+\tau_3+\tau_4)} \\ e^{-2(\tau_2+\tau_3+\tau_4)} (e^{-2\tau_3+\tau_4} + e^{-2\tau_1} (1 - e^{-\tau_4})) \end{bmatrix} \\
 &= \begin{bmatrix} y_3^2 y_4^2 \\ y_1^2 y_2^2 y_3^2 y_4^2 \\ y_1^2 y_2^2 y_3^2 y_4^2 \\ y_1^2 y_2^2 y_3^2 y_4^2 \\ 1 - y_4 (1 - y_1^2 y_2^2 y_3^2) \\ y_2^2 y_3^2 y_4^2 \\ y_2^2 y_3^2 y_4^2 (y_3^2 y_4 + y_1^2 (1 - y_4)) \end{bmatrix}.
 \end{aligned} \tag{6.73}$$

We form the ideal,

$$\begin{aligned}
 I = \langle & y_3^2 y_4^2 - q_{0011}, y_1^2 y_2^2 y_3^2 y_4^2 - q_{0101}, y_1^2 y_2^2 y_3^2 y_4^2 - q_{0110}, y_1^2 y_2^2 y_3^2 y_4^2 - q_{1001}, \\
 & 1 - y_4 (1 - y_1^2 y_2^2 y_3^2) - q_{1010}, y_2^2 y_3^2 y_4^2 - q_{1100}, y_2^2 y_3^2 y_4^2 (y_3^2 y_4 + y_1^2 (1 - y_4)) - q_{1111} \rangle.
 \end{aligned} \tag{6.74}$$

The Gröbner basis of the ideal is given in Appendix G, it has 29 generators. After finding the

Gröbner basis of the ideal, solving for the time parameters,

$$\begin{cases} y_1 &= \sqrt{\frac{q_{0101}}{q_{1100}}}, \\ y_2 &= \sqrt{\frac{q_{1100}}{q_{0011}}}, \\ y_3 &= \frac{\sqrt{q_{0011}} \left( -(1-q_{1010}) + \sqrt{(1-q_{1010})^2 + 4q_{0101}} \right)}{2q_{0101}}, \\ y_4 &= \frac{1}{2} \left( (1-q_{1010}) + \sqrt{(1-q_{1010})^2 + 4q_{0101}} \right). \end{cases} \quad (6.75)$$

Demanding  $y_1, y_2 \leq 1$  and from (6.37), (6.43) and (6.44), the constraints on Network 2b are then

$$\left\{ \begin{array}{ll} q_{0101} &= q_{0110}, \\ q_{0101} &= q_{1001}, \\ q_{0011}^2 q_{1100}^2 - 2q_{0011} q_{0101}^2 q_{1100} - q_{0011} q_{0101} q_{1010}^2 q_{1100} \\ &+ q_{0011} q_{0101} q_{1010} q_{1100} + q_{0011} q_{1010} q_{1100} q_{1111} - q_{0011} q_{1100} q_{1111} + q_{0101}^4 \\ &- q_{0101}^3 q_{1010} + q_{0101}^2 q_{1010} q_{1111} + q_{0101}^2 q_{1111} - q_{0101} q_{1111}^2 = 0, \\ q_{0101} &\leq q_{1010}, \\ q_{0101} &\leq q_{1100}, \\ q_{1100} &\leq q_{0011}, \\ (q_{0011} - q_{0101})^2 &\leq q_{0011} (1 - q_{1010})^2. \end{array} \right. \quad (6.76)$$

Networks 1b, 1c and 2b all have an equality constraint with 11 terms. These constraints look similar but are not equivalent. A constraint from one network cannot be transformed into a constraint from another network from a permutation on a transformed phylogenetic tensor element. For Network 1b the constraint is quartic, while for Networks 1c and 2b the constraint is quintic.

The constraints on the distances for Network 2b that are equivalent to the constraints on the distances for Network 2 are

$$d(3, 4) \leq d(1, 2) \leq d(2, 4) = d(2, 3) = d(1, 4). \quad (6.77)$$

Network 2b also has the distance constraint,  $d(1, 3) \leq d(2, 4)$ , while on Network 2,  $d(2, 4) = d(1, 3)$ . While on Network 2 the total network distance satisfies  $d(1, 2, 3, 4) = d(3, 4) + d(1, 2)$ , on Network 2b,  $d(1, 2, 3, 4) \neq d(3, 4) + d(1, 2)$ .

From  $d(3, 4) \leq d(2, 3)$  and  $d(1, 2) \leq d(1, 4)$  we can conclude that

$$d(3, 4) + d(1, 2) \leq d(2, 3) + d(1, 4). \quad (6.78)$$

From  $d(2, 4) = d(2, 3)$  and  $d(1, 3) \leq d(1, 4)$  we can conclude that

$$d(2, 4) + d(1, 3) \leq d(2, 3) + d(1, 4). \quad (6.79)$$

We can conclude that the sum of pairwise distances,  $d(2, 3) + d(1, 4)$ , must be greater than or equal to each of the other two sums of pairwise distances. For the four-point condition to be satisfied the two greatest sums of pairwise distances must be equal. We can conclude that Network 2b does not generally satisfy the four-point condition.

### 6.2.7 Network 3

The last tree or network that we will examine is the non-clock-like tree, which we will call Network 3, shown in Figure 6.11 below.

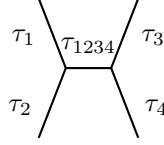


Figure 6.11: Network 3.

The variable transformed phylogenetic tensor elements are

$$\begin{bmatrix} q_{0011} \\ q_{0101} \\ q_{0110} \\ q_{1001} \\ q_{1010} \\ q_{1100} \\ q_{1111} \end{bmatrix} = \begin{bmatrix} e^{-(\tau_3 + \tau_4)} \\ e^{-(\tau_2 + \tau_4 + \tau_{1234})} \\ e^{-(\tau_2 + \tau_3 + \tau_{1234})} \\ e^{-(\tau_1 + \tau_4 + \tau_{1234})} \\ e^{-(\tau_1 + \tau_3 + \tau_{1234})} \\ e^{-(\tau_1 + \tau_2)} \\ e^{-(\tau_1 + \tau_2 + \tau_3 + \tau_4)} \end{bmatrix} = \begin{bmatrix} y_3 y_4 \\ y_2 y_4 y_{1234} \\ y_2 y_3 y_{1234} \\ y_1 y_4 y_{1234} \\ y_1 y_3 y_{1234} \\ y_1 y_2 \\ y_1 y_2 y_3 y_4 \end{bmatrix}. \quad (6.80)$$

We form the ideal,

$$\begin{aligned} I = \langle y_3 y_4 - q_{0011}, y_2 y_4 y_{1234} - q_{0101}, y_2 y_3 y_{1234} - q_{0110}, y_1 y_4 y_{1234} - q_{1001}, \\ y_1 y_3 y_{1234} - q_{1010}, y_1 y_2 - q_{1100}, y_1 y_2 y_3 y_4 - q_{1111} \rangle. \end{aligned} \quad (6.81)$$

The Gröbner basis of the ideal is given in Appendix H, it has 23 generators. After finding the



Gröbner basis of the ideal, solving for the time parameters,

$$\begin{cases} y_1 &= \sqrt{\frac{q_{1001}q_{1100}}{q_{0101}}}, \\ y_2 &= \sqrt{\frac{q_{0101}q_{1100}}{q_{1001}}}, \\ y_3 &= \sqrt{\frac{q_{0011}q_{0110}}{q_{0101}}}, \\ y_4 &= \sqrt{\frac{q_{0011}q_{0101}}{q_{0110}}}, \\ y_{1234} &= \sqrt{\frac{q_{0101}q_{0101}}{q_{0011}q_{1100}}}. \end{cases} \quad (6.82)$$

Demanding  $y_1, y_2 \leq 1$ , the constraints on the network are then

$$\begin{cases} q_{0101}q_{1010} &= q_{0110}q_{1001}, \\ q_{0011}q_{1100} &= q_{1111}, \\ q_{1001}q_{1100} &\leq q_{0101}, \\ q_{0101}q_{1100} &\leq q_{1001}, \\ q_{0011}q_{0110} &\leq q_{0101}, \\ q_{0011}q_{0101} &\leq q_{0110}, \\ q_{0101}q_{1010} &\leq q_{0011}q_{1100}. \end{cases} \quad (6.83)$$

The constraints on the distances for Network 3 are

$$\begin{cases} d(2,4) + d(1,3) &= d(2,3) + d(1,4), \\ d(3,4) + d(1,2) &= d(1,2,3,4), \\ d(2,4) &\leq d(1,4) + d(1,2), \\ d(1,4) &\leq d(2,4) + d(1,2), \\ d(2,4) &\leq d(3,4) + d(2,3), \\ d(2,3) &\leq d(3,4) + d(2,4), \\ d(3,4) + d(1,2) &\leq d(2,4) + d(1,3). \end{cases} \quad (6.84)$$

It can be shown that if the distance constraints for Network 1 or Network 2 are met then the distance constraints for Network 3 will also be met. This is to be expected since the clock-like trees can be recovered from the non-clock-like tree with the appropriate time parameter choices.

The second equality constraint is the constraint for the total network distance,  $d(3,4) + d(1,2) = d(1,2,3,4)$ . This is the same as for Networks 1, 1a and 2. From the last inequality constraint and the first equality constraint,

$$d(3,4) + d(1,2) \leq d(2,4) + d(1,3) = d(2,3) + d(1,4). \quad (6.85)$$

We can conclude that Network 3 satisfies the four-point condition, which is to be expected since

it is a tree.

### 6.3 Distinguishability of the Networks

To determine if one network is distinguishable from another we must compare the constraints on their transformed phylogenetic tensors. Suppose we have two sets,  $\Omega_1$  and  $\Omega_2$ , representing the probability spaces from the constraints for two networks. The intersection of two sets can have four options:

- (i)  $\Omega_1 = \Omega_2$ . The sets are equal,
  - (ii)  $\Omega_1 \neq \Omega_2$  and  $\Omega_1 \cap \Omega_2 = \Omega_1 \Leftrightarrow \Omega_1 \subset \Omega_2$ .  $\Omega_1$  is a proper subset of  $\Omega_2$ ,
  - (iii)  $\Omega_1 \cap \Omega_2 = \emptyset$ . The sets are disjoint,
  - (iv)  $\Omega_1 \neq \Omega_2$ ,  $\Omega_1 \cap \Omega_2 \subset \Omega_1$  and  $\Omega_1 \cap \Omega_2 \subset \Omega_2$ . The intersection of the sets is non-empty, but neither set is a proper subset of the other.
- (6.86)

In scenario (i) the two networks are indistinguishable from each other. In scenarios (ii), (iii) and (iv) the two networks are distinguishable from each other.

We will let  $\Omega_i$  denote the space of the constraints for Network  $i$ . We will start by examining the intersection of the constraints for Network 1 and Network 1a,  $\Omega_1 \cap \Omega_{1a}$ . Below in Table 6.1 is a comparison of the constraints for the two networks.

	Network	
	1	1a
Constraints	$q_{0101} = q_{0110}$	$q_{0101} = q_{0110}$
	$q_{1001} = q_{1010}$	$q_{1001} = q_{1010}$
	$q_{1001} = q_{1100}$	$q_{1001} \leq q_{1100}$
	$q_{0011}q_{1001} = q_{1111}$	$q_{0011}q_{1100} = q_{1111}$
	$q_{0101} \leq q_{0011}$	$q_{0101} \leq q_{0011}$
	$q_{1001} \leq q_{0101}$	$q_{1001} \leq q_{0101}$
		$(q_{0011} - q_{1001})^2 \leq q_{0011} (1 - q_{1100})^2$

Table 6.1: Summary of constraints for Network 1 and Network 1a.

Clearly the first two rows of constraints show identical equalities for both networks. We will consider the transformed phylogenetic tensor elements to be determined by others when there is an equality constraint. Consider  $q_{0110}$  to be determined by  $q_{0101}$  and  $q_{1010}$  to be determined by  $q_{1001}$ .  $q_{0110}$  and  $q_{1010}$  are not free elements and can be replaced by  $q_{0101}$  and  $q_{1001}$ , respectively. Likewise, for Network 1, but not for Network 1a, we consider  $q_{1100}$  to be determined by  $q_{1001}$ . In the intersection of the two sets of constraints  $q_{1100}$  will be determined by  $q_{1001}$ . For the constraints in the intersection of the two sets  $q_{1100}$  is not a free element and can be replaced by  $q_{1001}$ . The equalities in the fourth row of constraints must therefore be identical. We can see

that the inequalities in the fifth and sixth rows are identical. Hence, the constraints in Network 1a are the same constraints as in Network 1, with one extra inequality constraint. We will now look at the final constraint for Network 1a. Recognising that  $q_{1100}$  is not a free element, the last constraint on Network 1a becomes

$$(q_{0011} - q_{1001})^2 \leq q_{0011} (1 - q_{1001})^2. \quad (6.87)$$

Suppose we had an inequality in a similar form,

$$(q_a - q_b)^2 \leq q_a (1 - q_b)^2. \quad (6.88)$$

Then

$$\begin{aligned} & (q_a - q_b)^2 \leq q_a (1 - q_b)^2 \\ \Leftrightarrow & q_a^2 + q_b^2 - 2q_a q_b \leq q_a + q_a q_b^2 - 2q_a q_b \\ \Leftrightarrow & q_a^2 + q_b^2 \leq q_a + q_a q_b^2 \\ \Leftrightarrow & q_b^2 - q_a q_b^2 \leq q_a - q_a^2 \\ \Leftrightarrow & q_b^2 (1 - q_a) \leq q_a (1 - q_a). \end{aligned} \quad (6.89)$$

Since  $(1 - q_a) \neq 0$  in general,

$$q_b^2 (1 - q_a) \leq q_a (1 - q_a) \Leftrightarrow q_b^2 \leq q_a. \quad (6.90)$$

Since  $0 < q_b \leq 1$ ,

$$q_b^2 \leq q_b \leq q_a. \quad (6.91)$$

From the two inequality constraints,  $q_{1001} \leq q_{0101}$  and  $q_{0101} \leq q_{0011}$ , we can conclude that  $q_{1001} \leq q_{0011}$ . From (6.90) we can conclude that

$$(q_{0011} - q_{1001})^2 \leq q_{0011} (1 - q_{1001})^2. \quad (6.92)$$

Consequently, the last constraint on Network 1a can be ignored and the constraints in the

intersection of the probability spaces of Network 1 and Network 1a will be

$$\left\{ \begin{array}{l} q_{0101} = q_{0110}, \\ q_{1001} = q_{1010}, \\ q_{1001} = q_{1100}, \\ q_{0011}q_{1001} = q_{1111}, \\ q_{0101} \leq q_{0011}, \\ q_{1001} \leq q_{0101}. \end{array} \right. \quad (6.93)$$

This is the same set of constraints as Network 1. Hence,

$$\Omega_1 \cap \Omega_{1a} = \Omega_1. \quad (6.94)$$

It is interesting to note that the distances on Network 1a met the four-point condition, yet Network 1, the clock-like tree, and Network 1a are still distinguishable from each other. For the other intersections of the probability spaces for the other pairs of networks we used *Mathematica*. Below is an example of the *Mathematica* code we used.

```
(* All of the constraints on both Network i and Network j. Each
transformed phylogenetic tensor element must be in the interval,
(0,1]. ‘‘Reduce’’ removes any constraints that are implied by the
other constraints. *)
FullSimplify[Reduce[{Network i constraints, Network j constraints}, {
q0011, q0101, q0110, q1001, q1010, q1100, q1111}, Reals,
Backsubstitution -> True], 0 < q0011 <= 1 && 0 < q0101 <= 1 && 0 <
q0110 <= 1 && 0 < q1001 <= 1 && 0 < q1010 <= 1 && 0 < q1100 <= 1
&& 0 < q1111 <= 1]
```

For example, for the intersection of the probability spaces for Network 1 and Network 1a, we used the code below.

```
FullSimplify[Reduce[{q0101 == q0110, q1001 == q1010, q1001 == q1100,
q0011*q1001 == q1111, q0101 <= q0011, q1001 <= q0101, q0101 ==
q0110, q1001 == q1010, q0011*q1100 == q1111, q0101 <= q0011, q1001
<= q0101, q1001 <= q1100, (q0011 - q1001)^2 <= q0011*(1 - q1100)
^2}, {q0011, q0101, q0110, q1001, q1010, q1100, q1111}, Reals,
Backsubstitution -> True], 0 < q0011 <= 1 && 0 < q0101 <= 1 && 0 <
q0110 <= 1 && 0 < q1001 <= 1 && 0 < q1010 <= 1 && 0 < q1100 <= 1
&& 0 < q1111 <= 1]
```

Using *Mathematica* to find all of the intersections of the probability spaces for each pair of

networks, the intersections of the sets of constraints, denoted  $R$  with arbitrary subscripts, are

$$\left\{ \begin{array}{l} R_1 = \Omega_1 \cap \Omega_{1a} = \Omega_1 \cap \Omega_{1b} = \Omega_1 \cap \Omega_{1c} = \Omega_1 \cap \Omega_3 = \Omega_{1a} \cap \Omega_{1b} = \Omega_{1a} \cap \Omega_{1c} \\ \quad = \Omega_{1b} \cap \Omega_{1c} = \Omega_{1b} \cap \Omega_3 = \Omega_{1c} \cap \Omega_3 = \Omega_1, \\ R_2 = \Omega_1 \cap \Omega_2 = \Omega_1 \cap \Omega_{2b} = \Omega_{1b} \cap \Omega_2 = \Omega_{1b} \cap \Omega_{2b} = \Omega_{1c} \cap \Omega_2 \\ \quad = \Omega_{1c} \cap \Omega_{2b} \subset \Omega_1 \quad (\text{or } \subset \Omega_2), \\ R_3 = \Omega_{1a} \cap \Omega_3 \subset \Omega_{1a} \quad (\text{or } \subset \Omega_3), \\ R_4 = \Omega_{1a} \cap \Omega_2 = \Omega_{1a} \cap \Omega_{2b} \subset \Omega_2, \\ R_5 = \Omega_2 \cap \Omega_{2b} = \Omega_2 \cap \Omega_3 = \Omega_{2b} \cap \Omega_3 = \Omega_2. \end{array} \right. \quad (6.95)$$

From the first region we can conclude that  $\Omega_1$  is a proper subset of each of  $\Omega_{1a}$ ,  $\Omega_{1b}$ ,  $\Omega_{1c}$  and  $\Omega_3$ .  $\Omega_1$  must be a proper subset of each of  $\Omega_{1a}$ ,  $\Omega_{1b}$ ,  $\Omega_{1c}$  and  $\Omega_3$  and not equal to any of these probability spaces since each of the probability spaces has constraints that are not generally met on Network 1. This follows our intuition since the first structure of the clock-like tree, Network 1, can be reached from the convergence-divergence networks, Network 1a, Network 1b and Network 1c, or the non-clock-like tree, Network 3, by setting the appropriate time parameters to zero. The intersection of these five probability spaces will be  $\Omega_1$ . Summarising,

$$\left\{ \begin{array}{l} R_1 = \Omega_1 \cap \Omega_{1a} \cap \Omega_{1b} \cap \Omega_{1c} \cap \Omega_3 = \Omega_1, \\ R_1 = \Omega_1 \subset \Omega_{1a}, \\ R_1 = \Omega_1 \subset \Omega_{1b}, \\ R_1 = \Omega_1 \subset \Omega_{1c}, \\ R_1 = \Omega_1 \subset \Omega_3. \end{array} \right. \quad (6.96)$$

Likewise, from the fifth region we can conclude that  $\Omega_2$  is a proper subset of each of  $\Omega_{2b}$  and  $\Omega_3$ . This again follows our intuition since the second structure of the clock-like tree, Network 2, can be recovered from the convergence-divergence network, Network 2b, and the non-clock-like tree, Network 3, by setting the appropriate time parameters to zero. The intersection of the three probability spaces is  $\Omega_2$ . Summarising,

$$\left\{ \begin{array}{l} R_5 = \Omega_2 \cap \Omega_{2b} \cap \Omega_3 = \Omega_2, \\ R_5 = \Omega_2 \subset \Omega_{2b}, \\ R_5 = \Omega_2 \subset \Omega_3. \end{array} \right. \quad (6.97)$$

The intersections of the pairs of regions will be

$$\left\{ \begin{array}{l} R_1 \cap R_2 = \Omega_1 \cap (\Omega_1 \cap \Omega_2) = \Omega_1 \cap \Omega_2 = R_2, \\ R_1 \cap R_3 = \Omega_1 \cap (\Omega_{1a} \cap \Omega_3) = (\Omega_1 \cap \Omega_{1a}) \cap \Omega_3 = \Omega_1 \cap \Omega_3 = \Omega_1 = R_1, \\ R_1 \cap R_4 = \Omega_1 \cap (\Omega_{1a} \cap \Omega_2) = (\Omega_1 \cap \Omega_{1a}) \cap \Omega_2 = \Omega_1 \cap \Omega_2 = R_2, \\ R_1 \cap R_5 = \Omega_1 \cap \Omega_2 = R_2, \\ R_2 \cap R_3 = (\Omega_1 \cap \Omega_2) \cap (\Omega_{1a} \cap \Omega_3) = (\Omega_1 \cap \Omega_{1a}) \cap (\Omega_2 \cap \Omega_3) = \Omega_1 \cap (\Omega_2 \cap \Omega_3) \\ \quad = (\Omega_1 \cap \Omega_3) \cap \Omega_2 = \Omega_1 \cap \Omega_2 = R_2, \\ R_2 \cap R_4 = (\Omega_1 \cap \Omega_2) \cap (\Omega_{1a} \cap \Omega_2) = (\Omega_1 \cap \Omega_2) \cap \Omega_2 = \Omega_1 \cap \Omega_2 = R_2, \\ R_2 \cap R_5 = (\Omega_1 \cap \Omega_2) \cap \Omega_2 = \Omega_1 \cap \Omega_2 = R_2, \\ R_3 \cap R_4 = (\Omega_{1a} \cap \Omega_3) \cap (\Omega_{1a} \cap \Omega_2) = (\Omega_{1a} \cap \Omega_3) \cap \Omega_2 = \Omega_{1a} \cap (\Omega_2 \cap \Omega_3) \\ \quad = \Omega_{1a} \cap \Omega_2 = R_4, \\ R_3 \cap R_5 = (\Omega_{1a} \cap \Omega_3) \cap \Omega_2 = \Omega_{1a} \cap (\Omega_2 \cap \Omega_3) = \Omega_{1a} \cap \Omega_2 = R_4, \\ R_4 \cap R_5 = (\Omega_{1a} \cap \Omega_2) \cap \Omega_2 = \Omega_{1a} \cap \Omega_2 = R_4. \end{array} \right. \quad (6.98)$$

In summary,

$$\left\{ \begin{array}{l} R_1 \cap R_2 = R_2, \\ R_1 \cap R_3 = R_1, \\ R_1 \cap R_4 = R_2, \\ R_1 \cap R_5 = R_2, \\ R_2 \cap R_3 = R_2, \\ R_2 \cap R_4 = R_2, \\ R_2 \cap R_5 = R_2, \\ R_3 \cap R_4 = R_4, \\ R_3 \cap R_5 = R_4, \\ R_4 \cap R_5 = R_4. \end{array} \right. \quad (6.99)$$

From the first, fifth, sixth and seventh equations,

$$R_2 \subset R_1, R_2 \subset R_3, R_2 \subset R_4, R_2 \subset R_5. \quad (6.100)$$

We can conclude that in any region that  $R_2$  appears,  $R_1$ ,  $R_3$ ,  $R_4$  and  $R_5$  will also appear. This will be the region  $R_1 \cap R_2 \cap R_3 \cap R_4 \cap R_5$ . From the eighth, ninth and tenth equations,

$$R_4 \subset R_3, R_4 \subset R_5, R_3 \cap R_5 = R_4. \quad (6.101)$$

We can conclude that in any region that  $R_4$  appears,  $R_3$  and  $R_5$  will also appear. From the second, third and fourth equations,

$$R_1 \subset R_3, R_1 \cap R_4 = R_1 \cap R_5 = R_2. \quad (6.102)$$

We can conclude that in any region that  $R_1$  appears,  $R_3$  will also appear and the only region that will contain  $R_1$  and  $R_4$  or  $R_1$  and  $R_5$  will be the entire region that  $R_2$  appears in,  $R_1 \cap R_2 \cap R_3 \cap R_4 \cap R_5$ . There will also be the region where none of  $R_1, R_2, R_3, R_4$  or  $R_5$  will appear. There must therefore be six non-overlapping regions, each denoted  $S$ , with an arbitrary subscript. The six regions are

$$\begin{cases} S_1 &= R_1 \cap R_2 \cap R_3 \cap R_4 \cap R_5, \\ S_2 &= R_1 \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C, \\ S_3 &= R_1^C \cap R_2^C \cap R_3 \cap R_4 \cap R_5, \\ S_4 &= R_1^C \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C, \\ S_5 &= R_1^C \cap R_2^C \cap R_3^C \cap R_4^C \cap R_5, \\ S_6 &= R_1^C \cap R_2^C \cap R_3^C \cap R_4^C \cap R_5^C, \end{cases} \quad (6.103)$$

where  $C$  refers to the complement of the region. We will relabel the regions with the subscripts referring to the trees and networks that can be found in the region.  $S_6$  will be the region,

$$S_6 = S_{1a} \cup S_{1b} \cup S_{1c} \cup S_{2b} \cup S_3. \quad (6.104)$$

In other words, it is the region where only one tree or network can be found. There will be ten non-overlapping regions in total since  $S_6$  will have five non-overlapping regions, one for each tree or network in the region. See Appendix I for the working. These ten regions will be

$$\begin{cases} S_{1,1a,1b,1c,2,2b,3} &= \Omega_1 \cap \Omega_{1a} \cap \Omega_{1b} \cap \Omega_{1c} \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3, \\ S_{1,1a,1b,1c,3} &= \Omega_1 \cap \Omega_{1a} \cap \Omega_{1b} \cap \Omega_{1c} \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3, \\ S_{1a,2,2b,3} &= \Omega_1^C \cap \Omega_{1a} \cap \Omega_{1b}^C \cap \Omega_{1c}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3, \\ S_{1a,3} &= \Omega_1^C \cap \Omega_{1a} \cap \Omega_{1b}^C \cap \Omega_{1c}^C \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3, \\ S_{2,2b,3} &= \Omega_1^C \cap \Omega_{1a}^C \cap \Omega_{1b}^C \cap \Omega_{1c}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3, \\ S_{1a} &= \Omega_1^C \cap \Omega_{1a} \cap \Omega_{1b}^C \cap \Omega_{1c}^C \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3^C, \\ S_{1b} &= \Omega_1^C \cap \Omega_{1a}^C \cap \Omega_{1b} \cap \Omega_{1c}^C \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3^C, \\ S_{1c} &= \Omega_1^C \cap \Omega_{1a}^C \cap \Omega_{1b}^C \cap \Omega_{1c} \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3^C, \\ S_{2b} &= \Omega_1^C \cap \Omega_{1a}^C \cap \Omega_{1b}^C \cap \Omega_{1c}^C \cap \Omega_2^C \cap \Omega_{2b} \cap \Omega_3^C, \\ S_3 &= \Omega_1^C \cap \Omega_{1a}^C \cap \Omega_{1b}^C \cap \Omega_{1c}^C \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3. \end{cases} \quad (6.105)$$

Below in Figure 6.12 is a diagram representing the regions in the probability space. The region is in a two-dimensional space for convenience. The areas and shapes of each region are also inconsequential other than the reflection that some regions border each other in the probability space.

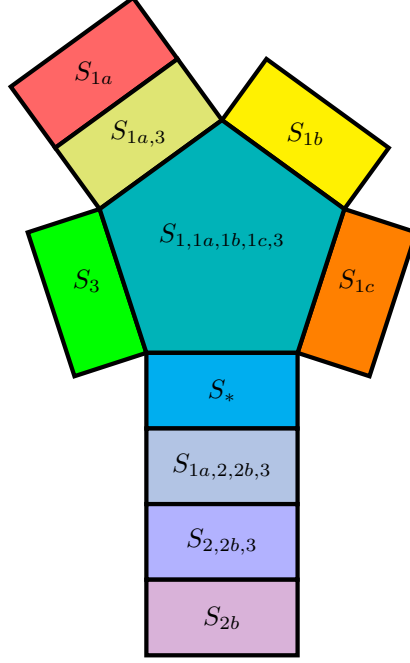


Figure 6.12: The probability space for every network. The subscripts refer to the networks that can be found in that region of the probability space.  $S_* = S_{1,1a,1b,1c,2,2b,3}$ .

Clearly each network is distinguishable from each other network since there are no two networks that share the same probability space despite there being some overlap in the probability spaces.

If a set of constraints could have arisen on a clock-like tree structure then it could have also arisen on any convergence-divergence network that is the clock-like tree structure with convergence at the leaves or the non-clock-like tree. This is to be expected since by setting the time parameter for convergence to zero we will recover the clock-like tree. Likewise, we can recover a clock-like tree from a non-clock-like tree by making the appropriate time parameters equal.

An interesting region is the region,  $S_{1a,3}$ , where the constraints could have arisen on a non-clock-like tree or a convergence-divergence network from the first clock-like tree structure, but no other trees or networks.

An important result is that we have regions where we can state that a set of constraints could only have arisen on a single convergence-divergence network or non-clock-like tree and no other trees or networks that we have examined. Interestingly, there is a region where the constraints



could have arisen on a convergence-divergence network from either tree structure, or the second clock-like tree structure or the non-clock-like tree. This is the region,  $S_{1a,2,2b,3}$ . The interesting result is that there is a region where the constraints could have arisen on the clock-like tree with one structure or on a convergence-divergence network with a different structure. Let's look at two of the networks in question, Network 1a and Network 2, as well as the other clock-like tree, Network 1. The three networks are shown in Figure 6.13 below.

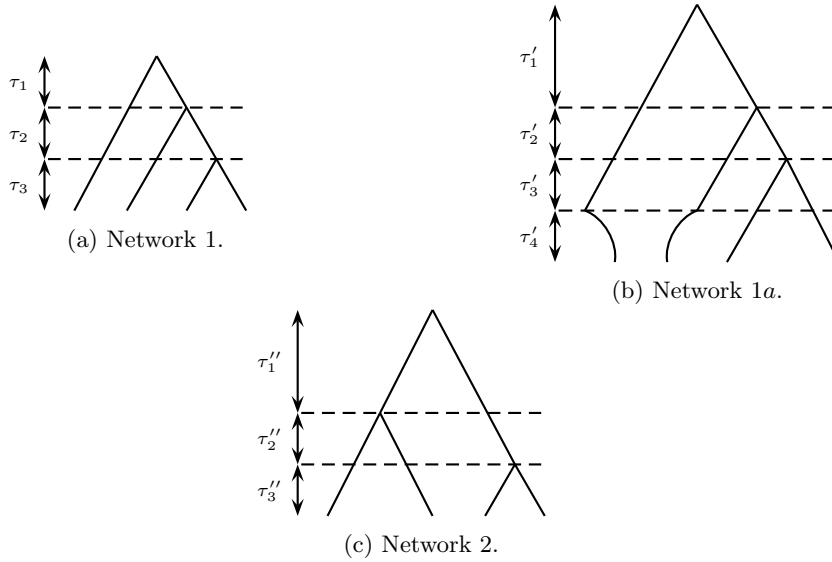


Figure 6.13: The three networks in question.

Recall from Definition 5.5 on page 102 that the *tree distance*,  $d(a, b)$ , is the sum of the edge lengths connecting two leaves,  $a$  and  $b$ . The tree distances on Network 1 will satisfy

$$d(3, 4) \leq d(2, 3) = d(2, 4) \leq d(1, 2) = d(1, 3) = d(1, 4). \quad (6.106)$$

The tree distances on Network 2 will satisfy

$$d(3, 4)'' \leq d(1, 2)'' \leq d(1, 3)'' = d(1, 4)'' = d(2, 3)'' = d(2, 4)'' . \quad (6.107)$$

Recall from Definition 5.5 on page 105 that the *pairwise distance* was used to define the distance between two leaves on a convergence-divergence network. Ignoring the distance between taxon 1 and taxon 2 on Network 1a for the time being, the distances on Network 1a will satisfy

$$d(3, 4)' \leq d(2, 3)' = d(2, 4)' \leq d(1, 3)' = d(1, 4)' . \quad (6.108)$$

Clearly if  $\tau'_4 = 0$  then  $d(1, 2)' = d(1, 3)' = d(1, 4)'$  and the distances on Network 1a will fit on Network 1. Now suppose that we let  $\tau'_4 \rightarrow \infty$ . The distance  $d(1, 2)'$  will become the smallest

and the distances will satisfy

$$d(1, 2)' \leq d(3, 4)' \leq d(2, 3)' = d(2, 4)' \leq d(1, 3)' = d(1, 4)'. \quad (6.109)$$

If  $\tau'_1 = 0$  then the distances become

$$d(1, 2)' \leq d(3, 4)' \leq d(2, 3)' = d(2, 4)' = d(1, 3)' = d(1, 4)'. \quad (6.110)$$

These distances will now fit on Network 2 if we permute taxa 1 and 3 and permute taxa 2 and 4.

Recall our discussion of clusters in Chapter 1. Labelling the leaves  $a, b, c, d$ , from left to right, the clusters on Network 1 will be

$$\left\{ \begin{array}{l} a, b, c, d, \\ b, c, d, \\ c, d, \\ a, \\ b, \\ c, \\ d. \end{array} \right. \quad (6.111)$$

The clusters on Network 2 will be

$$\left\{ \begin{array}{l} a, b, c, d, \\ a, b, \\ c, d, \\ a, \\ b, \\ c, \\ d. \end{array} \right. \quad (6.112)$$

The clusters on Network 1a are not immediately obvious. If  $\tau'_4 = 0$  then clearly Network 1a will be identical to Network 1 and they will have the same clusters. If  $\tau'_4$  is “large” then  $b$  will no longer cluster with  $c$  and  $d$ , but will instead cluster with  $a$ . The clusters would then be the same as those on Network 2. It therefore follows our intuition that there is a region where the constraints for both Network 1a and Network 2 are met.

### 6.3.1 Pseudocode Algorithm to Distinguish Trees and Networks

We can write an algorithm to determine which trees or networks a pattern frequency could have arisen on. We start by determining whether the constraints for an individual tree or network are met or not. In this algorithm we start with Network 1a, but we could have started with any network. We then determine which regions of the probability space the pattern frequency could lie in and which trees and networks are in that space. We then determine whether the constraints for a second tree or network are met or not and repeat the process. The algorithm terminates when we have reached one of the ten regions of the probability space in Figure 6.12 on page 154. Below is a pseudocode algorithm to find which trees or networks the pattern frequency could have arisen on.

```

Given some pattern frequency input,  $P$ ,
if Network 1a constraints == TRUE then
  if Network 1 constraints == TRUE then
    if Network 2 constraints == TRUE then
      tree or network  $\leftarrow$  Networks 1, 1a, 1b, 1c, 2, 2b or 3
    else
      tree or network  $\leftarrow$  Networks 1, 1a, 1b, 1c or 3
    end if
  else
    if Network 2 constraints == TRUE then
      tree or network  $\leftarrow$  Networks 1a, 2, 2b or 3
    else
      if Network 3 constraints == TRUE then
        tree or network  $\leftarrow$  Networks 1a or 3
      else
        tree or network  $\leftarrow$  Network 1a
      end if
    end if
  end if
else
  if Network 3 constraints == TRUE then
    if Network 2 constraints == TRUE then
      tree or network  $\leftarrow$  Networks 2, 2b or 3
    else
      tree or network  $\leftarrow$  Network 3
    end if
  else
    if Network 1b constraints == TRUE then
      tree or network  $\leftarrow$  Network 1b
    end if
  end if

```

```

else if Network 1c constraints == TRUE then
    tree or network  $\leftarrow$  Network 1c
else
    tree or network  $\leftarrow$  Network 2b
end if
end if
end if

```

We can determine which trees and networks a pattern frequency could have arisen on in either three or four steps. Given the restrictive nature of the binary symmetric model, it is possible that a pattern frequency could not have arisen on any of these trees or networks.

## 6.4 An Example with Multiple Convergence Periods

We have examined all of the four-taxon convergence-divergence trees and networks with at most a single convergence period, with the convergence period involving two leaves. We will now consider an example of a four-taxon convergence-divergence network with many convergence periods. This network is shown in Figure 6.14 below.

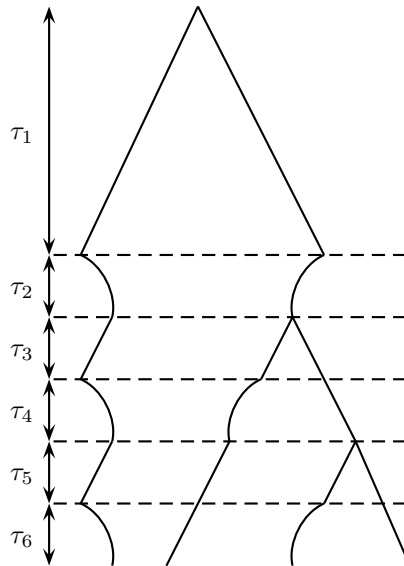


Figure 6.14: A four-taxon clock-like convergence-divergence network with multiple convergence periods.

It may be unlikely that this network would be biologically relevant. We know already that this convergence-divergence network will not be identifiable since it has a convergence period in one of the time periods when there are only two edges, nor will it be network identifiable or distinguishable from the convergence-divergence network that results when  $\tau_2 = 0$ . However,

it will highlight the power of our technique for analysing complicated networks. The variable transformed phylogenetic tensor elements are

$$\begin{aligned}
 \begin{bmatrix} q_{0011} \\ q_{0101} \\ q_{0110} \\ q_{1001} \\ q_{1010} \\ q_{1100} \\ q_{1111} \end{bmatrix} &= \begin{bmatrix} e^{-2(\tau_5+\tau_6)} \\ e^{-2(\tau_3+\tau_4+\tau_5+\tau_6)} \\ e^{-2(\tau_3+\tau_4+\tau_5+\tau_6)} \\ e^{-2(\tau_3+\tau_4+\tau_5+\tau_6)} (1 - e^{-\tau_2} (1 - e^{-2\tau_1})) \\ 1 - e^{-\tau_6} (1 - e^{-2(\tau_3+\tau_4+\tau_5)} (1 - e^{-\tau_2} (1 - e^{-2\tau_1}))) \\ e^{-2(\tau_5+\tau_6)} (1 - e^{-\tau_4} (1 - e^{-2\tau_3} (1 - e^{-\tau_2} (1 - e^{-2\tau_1})))) \\ e^{-2(\tau_5+\tau_6)} (e^{-2(\tau_3+\tau_4)} (1 - e^{-\tau_6}) + e^{-(2\tau_5+\tau_6)} (1 - e^{-\tau_4} (1 - e^{-2\tau_3} (1 - e^{-\tau_2} (1 - e^{-2\tau_1})))))) \end{bmatrix} \\
 &= \begin{bmatrix} y_5^2 y_6^2 \\ y_3^2 y_4^2 y_5^2 y_6^2 \\ y_3^2 y_4^2 y_5^2 y_6^2 \\ y_3^2 y_4^2 y_5^2 y_6^2 (1 - y_2 (1 - y_1^2)) \\ 1 - y_6 (1 - y_3^2 y_4^2 y_5^2 (1 - y_2 (1 - y_1^2))) \\ y_5^2 y_6^2 (1 - y_4 (1 - y_3^2 (1 - y_2 (1 - y_1^2)))) \\ y_5^2 y_6^2 (y_3^2 y_4^2 (1 - y_6) + y_5^2 y_6 (1 - y_4 (1 - y_3^2 (1 - y_2 (1 - y_1^2))))) \end{bmatrix}.
 \end{aligned} \tag{6.113}$$

We form the ideal,

$$\begin{aligned}
 I = \langle & y_5^2 y_6^2 - q_{0011}, y_3^2 y_4^2 y_5^2 y_6^2 - q_{0101}, y_3^2 y_4^2 y_5^2 y_6^2 - q_{0110}, y_3^2 y_4^2 y_5^2 y_6^2 (1 - y_2 (1 - y_1^2)) - q_{1001}, \\
 & 1 - y_6 (1 - y_3^2 y_4^2 y_5^2 (1 - y_2 (1 - y_1^2))) - q_{1010}, \\
 & y_5^2 y_6^2 (1 - y_4 (1 - y_3^2 (1 - y_2 (1 - y_1^2)))) - q_{1100}, \\
 & y_5^2 y_6^2 (y_3^2 y_4^2 (1 - y_6) + y_5^2 y_6 (1 - y_4 (1 - y_3^2 (1 - y_2 (1 - y_1^2))))) - q_{1111} \rangle.
 \end{aligned} \tag{6.114}$$

When we transform the basis of the ideal into the Gröbner basis we find that the ideal has 98 generators. Of these polynomial generators, the highest degree of the polynomials is 17 and the polynomial with the most monomials has 266 monomials. Clearly it is not practical to write down the Gröbner basis of the ideal here. We will not include the Gröbner basis of the ideal in the Appendix either. Instead, we will include the *Macaulay2* code for finding the Gröbner basis of the ideal, shown below.

```

S = QQ
R = S[y_1..y_6, q_1..q_7, MonomialOrder => {Lex => 6, Lex => 7}]
I = ideal(y_5^2*y_6^2 - q_1, y_3^2*y_4^2*y_5^2*y_6^2 - q_2,
y_3^2*y_4^2*y_5^2*y_6^2 - q_3,
y_3^2*y_4^2*y_5^2*y_6^2*(1 - y_2*(1 - y_1^2)) - q_4,
1 - y_6*(1 - y_3^2*y_4^2*y_5^2*(1 - y_2*(1 - y_1^2))) - q_5,

```

```

y_5^2*y_6^2*(1 - y_4*(1 - y_3^2*(1 - y_2*(1 - y_1^2)))) - q_6 ,
y_5^2*y_6^2*(y_3^2*y_4^2*(1 - y_6)
+ y_5^2*y_6*(1 - y_4*(1 - y_3^2*(1 - y_2*(1 - y_1^2)))) - q_7)
timing transpose gens gb I

```

The last line of code prints the Gröbner basis of the ideal and the CPU time. For this example, the CPU time was 0.249602 seconds.

Only two of the generators in the Gröbner basis do not contain any of the time parameters,  $y_i$ . Hence, there are only two equality constraints,

$$\left\{ \begin{array}{l} q_{0101} = q_{0110}, \\ q_{0011}^2 q_{1100}^2 - 2q_{0011} q_{0101} q_{1001} q_{1100} - q_{0011} q_{0101} q_{1010}^2 q_{1100} + q_{0011} q_{0101} q_{1010} q_{1100} \\ \quad + q_{0011} q_{1010} q_{1100} q_{1111} - q_{0011} q_{1100} q_{1111} + q_{0101}^2 q_{1010}^2 - q_{0101}^2 q_{1001} q_{1010} \\ \quad + q_{0101} q_{1001} q_{1010} q_{1111} + q_{0101} q_{1001} q_{1111} - q_{1001}^2 q_{1111}^2 = 0. \end{array} \right. \quad (6.115)$$

All other constraints will be inequalities that arise from the expressions for the time parameters and the constraints on these time parameters. Fortunately we do not need to know every generator of the Gröbner basis of the ideal to solve for the six time parameters. We only need six generators that we can form independent equations from that will allow us to solve for the six time parameters as functions of the transformed phylogenetic tensor elements.

If we look at the expressions for the transformed phylogenetic tensor elements in terms of the time parameters we can see that the time parameters,  $y_1$  and  $y_2$ , always appear in the same form. Every transformed phylogenetic tensor element expression containing these time parameters is of the form

$$q_i = f_1(y_3, y_4, y_5, y_6) + f_2(y_3, y_4, y_5, y_6) (1 - y_2 (1 - y_1^2)), \quad (6.116)$$

where  $f_1(y_3, y_4, y_5, y_6)$  and  $f_2(y_3, y_4, y_5, y_6)$  are expressions of the time parameters which do not include  $y_1$  or  $y_2$ .

Consequently,  $y_1$  and  $y_2$  will not be able to be recovered independently of each other and the network will not be identifiable. This is not particularly surprising since there is an obvious similarity in the structure of this network to the structure of the two-taxon convergence-divergence network introduced in Chapter 4. Both networks have two edges diverging from the root, which then converge back together. Recall that the two-taxon convergence-divergence network was not identifiable since the two time parameters, representing the divergence and convergence respectively, could not be recovered independently from the expressions for the transformed phylogenetic tensor elements.

We will use a set of five independent equations formed from five generators, some from the Gröbner basis and some from the original basis, to find expressions for the time parameters in

terms of the transformed phylogenetic tensor elements. There will be one expression involving the first two time parameters,  $y_1$  and  $y_2$ , since they cannot be recovered independently. One of these sets of five equations formed from five of the generators after some rearranging is

$$\left\{ \begin{array}{l} y_6^2 + q_{1010}y_6 - y_6 - q_{1001} = 0, \\ y_5^2 y_6^2 - q_{0011} = 0, \\ q_{0011}y_4^2 - q_{0011}y_4 + q_{1100}y_4 - q_{1001} = 0, \\ y_3^2 y_4^2 y_5^2 y_6^2 - q_{0101} = 0, \\ q_{0110}y_1^2 y_2 - q_{0110}y_2 + q_{0110} - q_{1001} = 0. \end{array} \right. \quad (6.117)$$

This set of equations was chosen because the polynomials in them had fewer terms than most of the other polynomials and because they contained fewer time parameters. For example, from the first equation it can easily be seen that an expression for  $y_6$  can be found since it contains none of the other time parameters.

A set of expressions for these time parameters is

$$\left\{ \begin{array}{l} 1 - y_2 (1 - y_1^2) = \frac{q_{1001}}{q_{0101}}, \\ y_3 = \frac{\sqrt{q_{0101}} \left( - (q_{0011} - q_{1100}) + \sqrt{(q_{0011} - q_{1100})^2 + 4q_{0011}q_{1001}} \right)}{2\sqrt{q_{0011}q_{1001}}}, \\ y_4 = \frac{(q_{0011} - q_{1100}) + \sqrt{(q_{0011} - q_{1100})^2 + 4q_{0011}q_{1001}}}{2q_{0011}}, \\ y_5 = \frac{\sqrt{q_{0011}} \left( - (1 - q_{1010}) + \sqrt{(1 - q_{1010})^2 + 4q_{1001}} \right)}{2q_{1001}}, \\ y_6 = \frac{1}{2} \left( (1 - q_{1010}) + \sqrt{(1 - q_{1010})^2 + 4q_{1001}} \right). \end{array} \right. \quad (6.118)$$

The inequality constraints on the transformed phylogenetic tensor elements can be found by demanding,  $0 < y_j \leq 1$ . Earlier in the thesis we established that if  $0 < y_1, y_2 \leq 1$  then  $0 < 1 - y_2 (1 - y_1^2) \leq 1$ . Demanding  $0 < y_3, y_4, y_5, y_6 \leq 1$  and  $0 < 1 - y_2 (1 - y_1^2) \leq 1$ , gives the constraints on the transformed phylogenetic tensor elements of

$$\left\{ \begin{array}{l} q_{1001} \leq q_{0101}, \\ q_{1001} \leq q_{1010}, \\ q_{1001} \leq q_{1100}, \\ \sqrt{q_{0011}} (q_{0101} - q_{1001}) \leq \sqrt{q_{0101}} (q_{0011} - q_{1100}), \\ (q_{0011} - q_{1001})^2 \leq q_{0011} (1 - q_{1010})^2. \end{array} \right. \quad (6.119)$$

Since  $q_{1001} \leq q_{0101}$ ,

$$\sqrt{q_{0011}}(q_{0101} - q_{1001}) \geq 0, \quad (6.120)$$

as required. However, since

$$\sqrt{q_{0011}}(q_{0101} - q_{1001}) \leq \sqrt{q_{0101}}(q_{0011} - q_{1100}), \quad (6.121)$$

it is implied that

$$q_{0011} - q_{1100} \geq 0, \quad (6.122)$$

and

$$q_{0011}(q_{0101} - q_{1001})^2 \leq q_{0101}(q_{0011} - q_{1100})^2. \quad (6.123)$$

We cannot take the square root of each side of this equality and preserve the inequality, however, since it must be assumed that  $q_{0011} - q_{1100} \geq 0$ . Consequently, the original sets of constraints is the set of constraints with the minimum number of elements in it. In conclusion, the set of constraints for this convergence-divergence network is

$$\left\{ \begin{array}{ll} q_{0101} & = q_{0110}, \\ q_{0011}^2 q_{1100}^2 - 2q_{0011} q_{0101} q_{1001} q_{1100} - q_{0011} q_{0101} q_{1010}^2 q_{1100} \\ + q_{0011} q_{0101} q_{1010} q_{1100} + q_{0011} q_{1010} q_{1100} q_{1111} - q_{0011} q_{1100} q_{1111} \\ + q_{0101}^2 q_{1010}^2 - q_{0101}^2 q_{1001} q_{1010} + q_{0101} q_{1001} q_{1010} q_{1111} \\ + q_{0101} q_{1001} q_{1111} - q_{1001} q_{1111}^2 & = 0, \\ q_{1001} & \leq q_{0101}, \\ q_{1001} & \leq q_{1010}, \\ q_{1001} & \leq q_{1100}, \\ \sqrt{q_{0011}}(q_{0101} - q_{1001}) & \leq \sqrt{q_{0101}}(q_{0011} - q_{1100}), \\ (q_{0011} - q_{1001})^2 & \leq q_{0011}(1 - q_{1010})^2. \end{array} \right. \quad (6.124)$$

We will not proceed to find the intersections between this set of constraints and the sets of constraints for other trees and networks since we have already argued that this network is not an appropriate choice for a network. However, these intersections could be found in the same way that we found the intersections in Section 6.3.



## Chapter 7

# Conclusion & Discussion

### 7.1 Thesis Findings

Sumner et al. [2012c] introduced a model for convergence on phylogenetic networks. We call these networks convergence-divergence networks. On these networks the assumption of independent evolution between edges is allowed to be broken and these edges converge with each other over a period of time. Some questions arise when we are attempting to determine whether the convergence-divergence networks are desirable choices for modelling. Three questions arise:

1. Are the trees and convergence-divergence networks *identifiable*?
2. Are the trees and convergence-divergence networks *network identifiable* from each other?
3. Are the trees and convergence-divergence networks *distinguishable* from each other?

For the convergence-divergence networks to be identifiable the numerical model parameters must be recoverable from the pattern frequencies. This is a matter of determining whether a set of solutions for the time parameters can be found from the expressions for the phylogenetic tensor for the network. For the convergence-divergence networks to be network identifiable the network itself must be recoverable from the pattern frequencies from all of the choices of trees and networks. In other words, can we find time parameters on one network that correspond to equivalent time parameters on another network? If we can do this then the convergence-divergence networks are not network identifiable. We used network identifiability to compare between taxon label permutations on a given tree or network. For any tree or network to be distinguishable from another tree or network there must be pattern frequencies that could have only arisen on one of the trees or networks. In other words, the probability spaces of the pattern frequencies for the two trees or networks must not be identical for them to be distinguishable.

In Chapter 4 and Chapter 6 we addressed the question of distinguishability by finding the equality and inequality constraints on the transformed phylogenetic tensors for various two-taxon, three-taxon and four-taxon trees and networks and comparing the intersections of the

probability spaces. A simple consequence of set theory is that the intersection of two sets,  $\Omega_1$  and  $\Omega_2$ , can take one of four possibilities:

- (i)  $\Omega_1 = \Omega_2$ . The sets are equal,
  - (ii)  $\Omega_1 \neq \Omega_2$  and  $\Omega_1 \cap \Omega_2 = \Omega_1 \Leftrightarrow \Omega_1 \subset \Omega_2$ .  $\Omega_1$  is a proper subset of  $\Omega_2$ ,
  - (iii)  $\Omega_1 \cap \Omega_2 = \emptyset$ . The sets are disjoint,
  - (iv)  $\Omega_1 \neq \Omega_2$ ,  $\Omega_1 \cap \Omega_2 \subset \Omega_1$  and  $\Omega_1 \cap \Omega_2 \subset \Omega_2$ . The intersection of the sets is non-empty, but neither set is a proper subset of the other.
- (7.1)

If  $\Omega_1$  and  $\Omega_2$  are taken to be two sets of constraints for two different trees or networks then we say that the two trees or networks are distinguishable unless  $\Omega_1 = \Omega_2$ .

All of the trees and networks that we examined for all number of taxa were modelled using the binary symmetric model since it is the simplest model to analyse. Being an Abelian group-based model it is diagonalisable, which allows us to find the constraints on the transformed phylogenetic tensors more easily. In Chapter 2 we examined a number of Abelian group-based models, some of which can be found in Sumner et al. [2014], and found the diagonalising matrices for these models.

We examined various two-taxon, three-taxon and four-taxon trees and networks. We made some restrictions on our convergence-divergence networks to simplify our analysis. For example, we did not consider any networks where there were alternating periods of convergence and divergence between the same group of taxa.

In Section 4.2 we looked at the distinguishability of two-taxon trees and networks. For the two-taxon case, there were three trees and networks to examine: the non-clock-like tree, the clock-like tree and the convergence divergence network with convergence between the two taxa. We found that the non-clock-like tree and the clock-like convergence-divergence network were not identifiable or distinguishable from the clock-like tree. We also found that the clock-like convergence-divergence network was not network identifiable from the clock-like tree. We concluded that under the binary symmetric model this model of convergence is not desirable over the clock-like tree for two taxa. We also showed that if a two-taxon convergence-divergence network is embedded inside a larger network then that network will not be identifiable, nor will it be network identifiable nor distinguishable from the same network with the convergence period replaced by a divergence period.

We then looked at the three-taxon case in Section 4.3, for which there were four trees and networks to examine: the clock-like tree, the non-clock-like tree, the convergence-divergence network with “sister” taxa convergence and the convergence-divergence network with “non-sister” taxa convergence. We are using the term “sister” to refer to edges in the same time period that form a cluster with no other edges contained in it and the term “non-sister” to refer to all other pairs of edges in the same time period. From the result regarding embedded two-taxon convergence-

divergence networks, we concluded that the three-taxon convergence-divergence network with sister taxa convergence was not identifiable, nor was it network identifiable nor distinguishable from the three-taxon clock-like tree. We found that the three-taxon convergence-divergence network with non-sister taxa convergence was identifiable and distinguishable from both the three-taxon non-clock-like tree and the three-taxon clock-like tree. We found that there were four regions in the probability space where the pattern frequencies could lie:

1. The pattern frequencies could lie on the convergence-divergence network with non-sister taxa convergence and on no other tree or network,
2. The pattern frequencies could lie on the non-clock-like tree and on no other tree or network,
3. The pattern frequencies could lie on either the convergence-divergence network with non-sister taxa convergence or the non-clock-like tree and on no other tree or network,
4. The pattern frequencies could lie on all three trees or networks: the convergence-divergence network with non-sister taxa convergence, the non-clock-like tree or the clock-like tree.

In Chapter 5 we examined the permutations of taxon labels for the three-taxon case. We first looked at the three-taxon clock-like tree. We showed that there are three taxon label permutations that are network identifiable from each other. Of the six possible taxon label permutations, there will be three pairs that have a taxon label permutation that is not network identifiable from the other taxon label permutation in the pair. Swapping the taxon labelling on the two sister taxa will not result in a network identifiable taxon label permutation from the original taxon label permutation.

We then looked at the convergence-divergence network with non-sister taxa convergence and compared the time parameters for different taxon label permutations. There was one taxon label permutation that was of particular interest to us, the permutation between the converging taxon not in the cluster of the two leaves sharing the most recent common ancestor and the non-converging taxon in that cluster. We found expressions for the time parameters on the second taxon label permutation in terms of the time parameters on the first taxon label permutation. Demanding the time parameters on the first taxon label permutation were positive, we found that for the transformed phylogenetic tensors to be equivalent one of the time parameters on the second taxon label permutation could be negative. From simulations we found that for 497 out of 500 samples all of the time parameters would be non-negative. For 3 of the 500 simulations one of the time parameters was negative. Since there are possible pattern frequencies that could have only arisen on one of the taxon label permutations, the two taxon label permutations are network identifiable. However, both taxon label permutations will usually have a set of non-negative time parameters that will give rise to the same phylogenetic tensors. There may therefore be a choice between the two taxon label permutations.

In Chapter 6 we looked at the distinguishability of four-taxon trees and networks. For four-taxon convergence-divergence networks the problem became far more complicated. We could

no longer find the constraints on the transformed phylogenetic tensors using elementary methods. We used methods from algebraic geometry to allow us to analyse these networks. As with the two-taxon and three-taxon cases, we first found the expressions for the transformed phylogenetic tensors for each tree and network. To find the equality and inequality constraints on the transformed phylogenetic tensors we first took the transformed phylogenetic tensor expressions and rearranged them to be identically zero. For example, suppose we had the transformed phylogenetic tensor expression,  $q_i = f(y_1, y_2, \dots, y_k)$ , where  $q_i$  is an element of the transformed phylogenetic tensor and  $y_i = e^{-\tau_i}$  are dependent on the time parameters. We then rearranged the expression to be  $f(y_1, y_2, \dots, y_k) - q_i = 0$ . We then took the expression,  $f(y_1, y_2, \dots, y_k) - q_i$ , to be a generating polynomial of an ideal. After finding all of the generators of the ideal we transformed the basis of the ideal into the Gröbner basis. The generators of the Gröbner basis were then extracted and formed into expressions that were identical to zero. These expressions contained all of the equality constraints on the transformed phylogenetic tensor. The time parameters could be solved from the remaining expressions. The inequality constraints on the transformed phylogenetic tensors came from the expressions for the time parameters and the fact that the time parameters must be non-negative on trees and networks.

In Section 6.3 we compared the constraints on each of the four-taxon trees and networks in the same fashion as for the two-taxon and three-taxon cases. We found that the two structures of the four-taxon clock-like trees, the four-taxon non-clock-like tree and all of the four-taxon convergence-divergence networks with convergence between two non-sister leaves were identifiable and distinguishable from each other. We concluded that all of these convergence-divergence networks may be appropriate choices for particular pattern frequencies.

Finally, we showed that we could find the constraints on a very complicated four-taxon convergence-divergence network with multiple convergence periods using our method. Despite the Gröbner basis having 98 generators, with the highest degree of the polynomials being 17 and the largest number of terms in the generators being 266, we were still able to find the set of constraints for this network.

We concluded that all three-taxon and four-taxon convergence-divergence networks with convergence between two non-sister leaves may be appropriate choices under the binary symmetric model when the pattern frequencies do not appear to have arisen on a tree. It must be noted, however, that our convergence-divergence networks may not always be an appropriate choice, even when it is clear that the pattern frequencies did not arise on a tree. If the pattern frequencies did not arise on a tree, then it cannot be assumed that the pattern frequencies arose on one of our convergence-divergence networks.

## 7.2 Future Work

We restricted our work to the binary symmetric model for simplicity. Being the simplest model, it is perhaps not surprising that the two-taxon convergence-divergence network was not identifiable,

network identifiable or distinguishable from the clock-like tree. For both the two-taxon clock-like tree and the two-taxon convergence-divergence network there was only a single variable element of the transformed phylogenetic tensor. Hence, any two-taxon network with multiple time parameters will not be identifiable under the binary symmetric model.

The work we have done on determining identifiability of trees and networks and distinguishability between one tree or network and another could be done for any Abelian group-based model. For more parameter rich Abelian group-based models there will be more polynomial equations to solve, with more variables, more terms and higher degrees, which will make the sets of constraints more difficult to analyse. A more parameter rich model or a model with more than two states will allow for more freedom for the two-taxon convergence-divergence network to be identifiable.

For more parameter rich models we have a choice of fixing the rate parameters to be identical across all time periods or allowing for variable rate parameters across the time periods. For the binary symmetric model this choice is not available. For only one rate parameter, different rates across time periods will be equivalent to equal rates across time periods, with a different set of time parameters. With more parameter rich models than the binary symmetric model it may be possible to distinguish the two-taxon convergence-divergence network from the two-taxon clock-like tree if rates are allowed to vary across time periods since this allows for more flexibility.

The increase in parameters may allow for a two-taxon convergence-divergence network that is identifiable, as well as network identifiable and distinguishable from the two-taxon clock-like tree.

We didn't analyse any trees or networks beyond the four-taxon case, however there is no reason not to examine trees and networks with more taxa. If the trend from the three-taxon and four-taxon convergence-divergence networks holds for more taxa then any  $n$ -taxon convergence-divergence networks with convergence between two non-sister leaves will be distinguishable from  $n$ -taxon clock-like trees and  $n$ -taxon non-clock-like trees.

We only examined the taxon label permutations for the three-taxon case. It would be interesting to see if there are any taxon label permutations for the four-taxon or larger taxon cases which are not network identifiable from each other.

The least parameter rich of the trees and networks are the clock-like trees. Steel [2005] noted that when developing models care must be taken to avoid overfitting the data. The more parameters that a tree or network has the more potential there is to overfit the data. Steel [2005] also noted, however, that the models must be biologically relevant. In some circumstances a clock-like tree will not be biologically relevant and will not be the best fit to the data. If there is still variability in the data that is not explained by a clock-like tree then we can introduce more parameters to the model, provided these extra parameters significantly improve the fit. One option to introduce more parameters is to remove the molecular clock assumption and allow for a non-clock-like tree. Another option is to remove the assumption of independent evolution across edges in the same time period, allowing for convergence. Non-clock-like trees and convergence-divergence networks do not necessarily have the same number of parameters. Below in Table 7.1

is the number of parameters for clock-like trees, non-clock-like trees and convergence-divergence networks for different number of taxa.

Number of taxa	Number of parameters		
	Clock-like tree	Non-clock-like tree	Convergence-divergence network
2	1	-	$\geq 2$
3	2	3	$\geq 3$
4	3	5	$\geq 4$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$n$	$n - 1$	$2n - 3$	$\geq n$

Table 7.1: Number of parameters for clock-like trees, non-clock-like trees and clock-like convergence-divergence networks.

The number of parameters for an  $n$ -taxon clock-like tree is  $n - 1$ . If we wish to remove the molecular clock assumption, an  $n$ -taxon non-clock-like tree will have  $2n - 3$  parameters,  $2n - 3 - (n - 1) = n - 2$  parameters more than the clock-like tree. If we have a large number of taxa then removing the molecular clock assumption introduces many more parameters, approximately doubling the number of parameters. Clearly this is a potential issue of overfitting. With convergence-divergence networks we have much more flexibility in our parameter number choice. Introducing one convergence period to a clock-like tree will only introduce one extra parameter. We have the option of introducing multiple convergence periods if there is still more variability in the data left to explain. Convergence-divergence networks could be an option when there is more variability in the data than can be explained by a clock-like tree, but when the data is overfit on a non-clock-like tree.

Suppose we decided on a convergence-divergence network, however we did not decide on where the taxa should be placed on the leaves. Recall from Chapter 5 and earlier in Section 7.1 that for the three-taxon convergence-divergence network with convergence between non-sister taxa all of the taxon label permutations were network identifiable. However, of the six taxon labellings, there were three pairs of permutations that almost always gave rise to equivalent transformed phylogenetic tensors with all parameters being non-negative. We found that for any given taxon labelling, if we swapped the labels between the converging edge having the root as the most recent common ancestor with the other two leaves and the edge which is not converging with another edge then the two taxon label permutations would almost always give rise to equivalent transformed phylogenetic tensors with all parameters being non-negative. Two of those taxon labellings are shown in Figure 7.1 below.

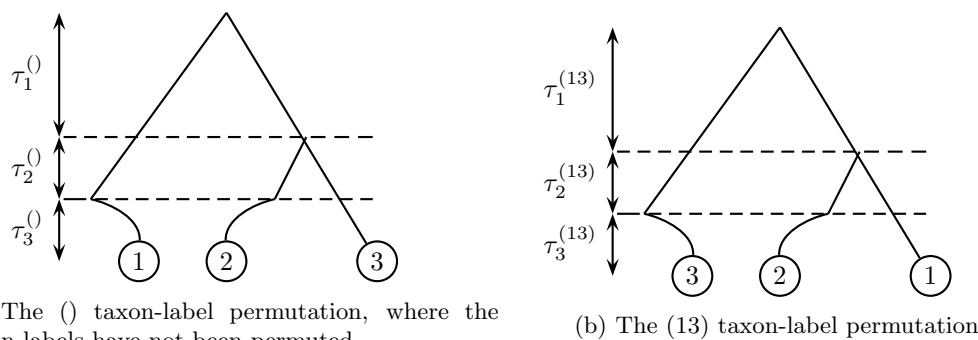


Figure 7.1: The two taxon-label permutations in question for the three-taxon convergence-divergence network.

If we take  $\tau_1^{()}$ ,  $\tau_2^{()}$  and  $\tau_3^{()}$  to all be non-negative then  $\tau_1^{(13)}$  and  $\tau_3^{(13)}$  must always be non-negative, however  $\tau_2^{(13)}$  will occasionally be negative. In most cases  $\tau_2^{(13)}$  will also be non-negative, however. In these circumstances we will have a choice between the two taxon label permutations. This choice could be made by considering the biological implications of the two taxon label permutations. If it is known, or suspected, that a pair of taxa converged then we could choose the taxon label permutation that allowed that pair of taxa to converge. We could also choose the taxon label permutation that has the least convergence. In other words, we could compare  $\tau_3^{()}$  and  $\tau_3^{(13)}$  and choose the taxon label permutation that had the smaller of the two parameters. Another choice that could be made is choosing the taxon label permutation that converges the two taxa with the least divergence or distance between them. In other words, we could compare  $\tau_1^{()} + \tau_2^{()}$  to  $\tau_1^{(13)} + \tau_2^{(13)}$  and choose the taxon label permutation with the smaller of the two sums.

It would be interesting to apply the results on identifiability and distinguishability of three-taxon and four-taxon trees and networks to some data sets for which a clock-like tree does not provide enough parameters and a non-clock-like tree provides too many parameters. The questions of identifiability and distinguishability could be used as a diagnostic tool for given data sets, particularly data sets for which there is known to be or suspected to be some form of convergence. An example for which convergence-divergence networks under the binary symmetric model could be used as a diagnostic tool is binary single nucleotide polymorphisms (SNPS).

Holland et al. [2010] analysed a binary morphological data set of cormorants and shags, with 30 taxa and 137 morphological characters. The taxa have been previously classified on morphological characteristics, which is reflected in their genus names and morphological consensus tree, with the nine genera representing the nine distinct morphological groupings. They showed that the molecular consensus tree differed considerably from the morphological consensus tree. The convergence-divergence networks could be applied to the morphological data set to further address the question of whether the species developed similar morphological traits due to inhabiting similar niches rather than sharing a recent common ancestor.

Finally, we have shown that the convergence-divergence networks introduced by Sumner et al. [2012c] are a useful new way to model convergence. We have shown that these networks could be more appropriate models than trees when a pattern frequency does not appear to have arisen on a tree. These convergence-divergence networks could be used as a diagnostic tool when it is suspected that a set of taxa did not arise on a tree, with convergence having occurred between some of the taxa.



# Appendices



## Appendix A

# Time Parameters for the (12) Taxon Label Permutation

Equating the three pairwise distances for the two taxon label permutations for the three-taxon convergence-divergence network, the (12) taxon label permutation and no taxon label permutation,

$$\left\{ \begin{array}{l} 2 \left( \tau_2^{(0)} + \tau_3^{(0)} \right) = 2 \left( \tau_1^{(12)} + \tau_2^{(12)} + \tau_3^{(12)} \right), \\ 2 \left( \tau_1^{(0)} + \tau_2^{(0)} + \tau_3^{(0)} \right) = 2 \left( \tau_2^{(12)} + \tau_3^{(12)} \right), \\ -\ln \left( 1 - e^{-\tau_3^{(0)}} \left( 1 - e^{-2(\tau_1^{(0)} + \tau_2^{(0)})} \right) \right) = -\ln \left( 1 - e^{-\tau_3^{(12)}} \left( 1 - e^{-2(\tau_1^{(12)} + \tau_2^{(12)})} \right) \right). \end{array} \right. \quad (\text{A.1})$$

Making the substitutions,  $x_i = e^{-\tau_i^{(0)}}$  and  $y_i = e^{-\tau_i^{(12)}}$ , we get the polynomial equations,

$$\left\{ \begin{array}{l} x_2^2 x_3^2 = y_1^2 y_2^2 y_3^2, \\ x_1^2 x_2^2 x_3^2 = y_2^2 y_3^2, \\ 1 - x_3 (1 - x_1^2 x_2^2) = 1 - y_3 (1 - y_1^2 y_2^2). \end{array} \right. \quad (\text{A.2})$$

Making the further substitutions,  $v_1 = x_1^2$ ,  $v_2 = x_2^2 x_3$ ,  $v_3 = x_3$ ,  $w_1 = y_1^2$ ,  $w_2 = y_2^2 y_3$ ,  $w_3 = y_3$ ,

$$\left\{ \begin{array}{l} v_2 v_3 = w_1 w_2 w_3, \\ v_1 v_2 v_3 = w_2 w_3, \\ 1 - v_3 + v_1 v_2 = 1 - w_3 + w_1 w_2. \end{array} \right. \quad (\text{A.3})$$

Making the further substitutions,  $v_2v_3 = \alpha$ ,  $v_1v_2v_3 = \beta$ , and  $1 - v_3 + v_1v_2 = \gamma$ ,

$$\begin{cases} \alpha &= w_1w_2w_3, \\ \beta &= w_2w_3, \\ \gamma &= 1 - w_3 + w_1w_2. \end{cases} \quad (\text{A.4})$$

We will now solve the system for the variables,  $w_1$ ,  $w_2$  and  $w_3$ , in terms of the variables,  $\alpha$ ,  $\beta$  and  $\gamma$ . We can immediately see that dividing the first equation by the second equation gives

$$w_1 = \frac{\alpha}{\beta}. \quad (\text{A.5})$$

Now rearranging the second equation to solve for  $w_2$ ,

$$w_2 = \frac{\beta}{w_3}. \quad (\text{A.6})$$

Substituting in the expressions for  $w_1$  and  $w_2$  into the third equation,

$$\begin{aligned} \gamma &= 1 - w_3 + \frac{\alpha}{\beta} \frac{\beta}{w_3} \\ \Leftrightarrow \gamma &= 1 - w_3 + \frac{\alpha}{w_3} \\ \Leftrightarrow -(1 - \gamma) &= -w_3 + \frac{\alpha}{w_3} \\ \Leftrightarrow w_3 - (1 - \gamma) - \frac{\alpha}{w_3} &= 0 \\ \Leftrightarrow w_3^2 - (1 - \gamma)w_3 - \alpha &= 0 \\ \Leftrightarrow w_3 &= \frac{1}{2} \left( (1 - \gamma) \pm \sqrt{(1 - \gamma)^2 + 4\alpha} \right). \end{aligned} \quad (\text{A.7})$$

Demanding  $w_3 = y_3 = e^{-\tau_3^{(12)}} > 0$ , the only possible solution is

$$w_3 = \frac{1}{2} \left( (1 - \gamma) + \sqrt{(1 - \gamma)^2 + 4\alpha} \right). \quad (\text{A.8})$$

We can now substitute in the expression for  $w_3$  to find the expression for  $w_2$ ,

$$\begin{aligned}
w_2 &= \frac{\beta}{w_3} \\
&= \frac{\beta}{\frac{1}{2} \left( (1 - \gamma) + \sqrt{(1 - \gamma)^2 + 4\alpha} \right)} \\
&= \frac{2\beta}{(1 - \gamma) + \sqrt{(1 - \gamma)^2 + 4\alpha}} \\
&= \frac{2\beta}{(1 - \gamma) + \sqrt{(1 - \gamma)^2 + 4\alpha}} \cdot \frac{-(1 - \gamma) + \sqrt{(1 - \gamma)^2 + 4\alpha}}{-(1 - \gamma) + \sqrt{(1 - \gamma)^2 + 4\alpha}} \\
&= \frac{2\beta \left( -(1 - \gamma) + \sqrt{(1 - \gamma)^2 + 4\alpha} \right)}{4\alpha} \\
&= \frac{\beta}{2\alpha} \left( -(1 - \gamma) + \sqrt{(1 - \gamma)^2 + 4\alpha} \right).
\end{aligned} \tag{A.9}$$

We now have the expressions for  $w_1$ ,  $w_2$  and  $w_3$ . Recall that

$$w_1 = y_1^2 = e^{-2\tau_1^{(12)}} \tag{A.10}$$

and

$$w_1 = \frac{\alpha}{\beta}. \tag{A.11}$$

We can now use the appropriate back-substitutions to get expressions for the time parameters on the convergence-divergence network with the (12) taxon label permutation in terms of the time parameters on the convergence-divergence network with no taxon label permutation.

The first time parameter will be

$$\begin{aligned}
 \tau_1^{(12)} &= -\frac{1}{2} \ln \left( \frac{\alpha}{\beta} \right) \\
 &= \ln \sqrt{\frac{\beta}{\alpha}} \\
 &= \ln \sqrt{\frac{v_1 v_2 v_3}{v_2 v_3}} \\
 &= \ln \sqrt{v_1} \\
 &= \ln \sqrt{x_1^2} \\
 &= \ln (x_1) \\
 &= \ln \left( e^{-\tau_1^{(0)}} \right) \\
 &= -\tau_1^{(0)}.
 \end{aligned} \tag{A.12}$$

Recall that

$$y_2^2 = e^{-2\tau_2^{(12)}}. \tag{A.13}$$

Recall that

$$w_2 = \frac{\beta}{w_3} = y_2^2 y_3 \tag{A.14}$$

and

$$w_3 = y_3. \tag{A.15}$$

Consequently,

$$\begin{aligned}
 y_2^2 &= \frac{w_2}{w_3} \\
 &= \frac{\beta}{w_3^2} \\
 &= \frac{4\beta}{\left( (1-\gamma) + \sqrt{(1-\gamma)^2 + 4\alpha} \right)^2}.
 \end{aligned} \tag{A.16}$$

Again, using the appropriate substitutions,

$$\begin{aligned}
\tau_2^{(12)} &= -\frac{1}{2} \ln \left( \frac{4\beta}{\left( (1-\gamma) + \sqrt{(1-\gamma)^2 + 4\alpha} \right)^2} \right) \\
&= \ln \left( \frac{(1-\gamma) + \sqrt{(1-\gamma)^2 + 4\alpha}}{2\sqrt{\beta}} \right) \\
&= \ln \left( \frac{(1 - (1 - v_3 + v_1 v_2)) + \sqrt{(1 - (1 - v_3 + v_1 v_2))^2 + 4v_2 v_3}}{2\sqrt{v_1 v_2 v_3}} \right) \\
&= \ln \left( \frac{(v_3 - v_1 v_2) + \sqrt{(v_3 - v_1 v_2)^2 + 4v_2 v_3}}{2\sqrt{v_1 v_2 v_3}} \right) \\
&= \ln \left( \frac{(x_3 - x_1^2 x_2^2 x_3) + \sqrt{(x_3 - x_1^2 x_2^2 x_3)^2 + 4x_2^2 x_3^2}}{2\sqrt{x_1^2 x_2^2 x_3^2}} \right) \\
&= \ln \left( \frac{x_3 (1 - x_1^2 x_2^2) + \sqrt{x_3^2 (1 - x_1^2 x_2^2)^2 + 4x_2^2 x_3^2}}{2x_1 x_2 x_3} \right) \tag{A.17} \\
&= \ln \left( \frac{x_3 (1 - x_1^2 x_2^2) + x_3 \sqrt{(1 - x_1^2 x_2^2)^2 + 4x_2^2}}{2x_1 x_2 x_3} \right) \\
&= \ln \left( \frac{x_3 \left( (1 - x_1^2 x_2^2) + \sqrt{(1 - x_1^2 x_2^2)^2 + 4x_2^2} \right)}{2x_1 x_2 x_3} \right) \\
&= \ln \left( \frac{(1 - x_1^2 x_2^2) + \sqrt{(1 - x_1^2 x_2^2)^2 + 4x_2^2}}{2x_1 x_2} \right) \\
&= \ln \left( \frac{\left( 1 - e^{-2(\tau_1^0 + \tau_2^0)} \right) + \sqrt{\left( 1 - e^{-2(\tau_1^0 + \tau_2^0)} \right)^2 + 4e^{-2\tau_2^0}}}{2e^{-(\tau_1^0 + \tau_2^0)}} \right).
\end{aligned}$$

Recall that

$$w_3 = y_3 = e^{-\tau_3^{(12)}} \tag{A.18}$$

and

$$w_3 = \frac{1}{2} \left( (1 - \gamma) + \sqrt{(1 - \gamma)^2 + 4\alpha} \right). \quad (\text{A.19})$$



Again, using the appropriate substitutions,

$$\begin{aligned}
\tau_3^{(12)} &= -\ln \left( \frac{1}{2} \left( (1-\gamma) + \sqrt{(1-\gamma)^2 + 4\alpha} \right) \right) \\
&= \ln \left( \frac{2}{(1-\gamma) + \sqrt{(1-\gamma)^2 + 4\alpha}} \right) \\
&= \ln \left( \frac{2}{(1-\gamma) + \sqrt{(1-\gamma)^2 + 4\alpha}} \cdot \frac{-(1-\gamma) + \sqrt{(1-\gamma)^2 + 4\alpha}}{-(1-\gamma) + \sqrt{(1-\gamma)^2 + 4\alpha}} \right) \\
&= \ln \left( \frac{2 \left( -(1-\gamma) + \sqrt{(1-\gamma)^2 + 4\alpha} \right)}{4\alpha} \right) \\
&= \ln \left( \frac{-(1-\gamma) + \sqrt{(1-\gamma)^2 + 4\alpha}}{2\alpha} \right) \\
&= \ln \left( \frac{-(1 - (1 - v_3 + v_1 v_2)) + \sqrt{(1 - (1 - v_3 + v_1 v_2))^2 + 4v_2 v_3}}{2v_2 v_3} \right) \\
&= \ln \left( \frac{-(v_3 - v_1 v_2) + \sqrt{(v_3 - v_1 v_2)^2 + 4v_2 v_3}}{2v_2 v_3} \right) \\
&= \ln \left( \frac{-(x_3 - x_1^2 x_2^2 x_3) + \sqrt{(x_3 - x_1^2 x_2^2 x_3)^2 + 4x_2^2 x_3^2}}{2x_2^2 x_3^2} \right) \\
&= \ln \left( \frac{-x_3 (1 - x_1^2 x_2^2) + \sqrt{x_3^2 (1 - x_1^2 x_2^2)^2 + 4x_2^2 x_3^2}}{2x_2^2 x_3^2} \right) \\
&= \ln \left( \frac{-x_3 (1 - x_1^2 x_2^2) + x_3 \sqrt{(1 - x_1^2 x_2^2)^2 + 4x_2^2}}{2x_2^2 x_3^2} \right) \\
&= \ln \left( \frac{x_3 \left( -(1 - x_1^2 x_2^2) + \sqrt{(1 - x_1^2 x_2^2)^2 + 4x_2^2} \right)}{2x_2^2 x_3^2} \right) \\
&= \ln \left( \frac{-(1 - x_1^2 x_2^2) + \sqrt{(1 - x_1^2 x_2^2)^2 + 4x_2^2}}{2x_2^2 x_3} \right) \\
&= \ln \left( \frac{-\left(1 - e^{-2(\tau_1^{(0)} + \tau_2^{(0)})}\right) + \sqrt{\left(1 - e^{-2(\tau_1^{(0)} + \tau_2^{(0)})}\right)^2 + 4e^{-2\tau_2^{(0)}}}}{2e^{-(2\tau_2^{(0)} + \tau_3^{(0)})}} \right).
\end{aligned} \tag{A.20}$$

In summary,

$$\begin{cases} \tau_1^{(12)} &= -\tau_1^{(0)}, \\ \tau_2^{(12)} &= \ln \left( \frac{\left(1 - e^{-2(\tau_1^{(0)} + \tau_2^{(0)})}\right) + \sqrt{\left(1 - e^{-2(\tau_1^{(0)} + \tau_2^{(0)})}\right)^2 + 4e^{-2\tau_2^{(0)}}}}{2e^{-(\tau_1^{(0)} + \tau_2^{(0)})}} \right), \\ \tau_3^{(12)} &= \ln \left( \frac{-\left(1 - e^{-2(\tau_1^{(0)} + \tau_2^{(0)})}\right) + \sqrt{\left(1 - e^{-2(\tau_1^{(0)} + \tau_2^{(0)})}\right)^2 + 4e^{-2\tau_2^{(0)}}}}{2e^{-(2\tau_2^{(0)} + \tau_3^{(0)})}} \right). \end{cases} \quad (\text{A.21})$$

Recall that  $\tau_2^{(12)}$  can be expressed as

$$\tau_2^{(12)} = \ln \left( \frac{(1 - x_1^2 x_2^2) + \sqrt{(1 - x_1^2 x_2^2)^2 + 4x_2^2}}{2x_1 x_2} \right). \quad (\text{A.22})$$

Recall that  $x_i = e^{-\tau_i^{(0)}}$ . Demanding  $0 < x_1, x_2 \leq 1$ ,

$$\begin{aligned} (1 - x_1^2 x_2^2) + \sqrt{(1 - x_1^2 x_2^2)^2 + 4x_2^2} &\geq \sqrt{(1 - x_1^2 x_2^2)^2 + 4x_2^2} \\ &\geq \sqrt{4x_2^2} \\ &= 2x_2 \\ &\geq 2x_1 x_2. \end{aligned} \quad (\text{A.23})$$

We can conclude that  $\tau_2^{(12)} \geq 0$ .

Recall that  $\tau_3^{(12)}$  can be expressed as

$$\tau_3^{(12)} = \ln \left( \frac{-(1 - x_1^2 x_2^2) + \sqrt{(1 - x_1^2 x_2^2)^2 + 4x_2^2}}{2x_2^2 x_3} \right). \quad (\text{A.24})$$

Demanding  $0 < x_1, x_2, x_3 \leq 1$  and  $\tau_3^{(12)} \geq 0$ ,

$$\begin{aligned}
& - (1 - x_1^2 x_2^2) + \sqrt{(1 - x_1^2 x_2^2)^2 + 4x_2^2} \geq 2x_2^2 x_3 \\
& \Leftrightarrow \sqrt{(1 - x_1^2 x_2^2)^2 + 4x_2^2} \geq (1 - x_1^2 x_2^2) + 2x_2^2 x_3 \\
& \Leftrightarrow (1 - x_1^2 x_2^2)^2 + 4x_2^2 \geq (1 - x_1^2 x_2^2)^2 + 4x_2^4 x_3^2 + 4x_2^2 x_3 (1 - x_1^2 x_2^2) \\
& \Leftrightarrow 4x_2^2 \geq 4x_2^4 x_3^2 + 4x_2^2 x_3 (1 - x_1^2 x_2^2) \\
& \Leftrightarrow 1 \geq x_2^2 x_3^2 + x_3 (1 - x_1^2 x_2^2) \\
& \Leftrightarrow 1 - x_3 (1 - x_1^2 x_2^2) - x_2^2 x_3^2 \geq 0. \\
& \Leftrightarrow 1 - x_3 (1 - x_1^2 x_2^2 + x_2^2 x_3) \geq 0. \\
& \Leftrightarrow 1 - x_3 (1 - x_2^2 (x_1^2 - x_3)) \geq 0.
\end{aligned} \tag{A.25}$$

It is not immediately clear whether this inequality is true and  $\tau_3^{(12)} \geq 0$ . Suppose  $x_3 < x_1^2$ . Then

$$\begin{aligned}
& 0 < x_1^2 - x_3 \\
& \Leftrightarrow 0 < x_2^2 (x_1^2 - x_3) \\
& \Leftrightarrow -x_2^2 (x_1^2 - x_3) < 0 \\
& \Leftrightarrow 1 - x_2^2 (x_1^2 - x_3) < 1 \\
& \Leftrightarrow x_3 (1 - x_2^2 (x_1^2 - x_3)) < 1 \\
& \Leftrightarrow -1 < -x_3 (1 - x_2^2 (x_1^2 - x_3)) \\
& \Leftrightarrow 0 < 1 - x_3 (1 - x_2^2 (x_1^2 - x_3)).
\end{aligned} \tag{A.26}$$

We can conclude that the inequality will be satisfied when  $x_3 < x_1^2$ . Clearly, when  $x_3 = x_1^2$  the inequality will be satisfied, since  $1 - x_3 > 0$ . We can conclude that  $\tau_3^{(12)}$  will sometimes be non-negative and sometimes be negative.

To get a rough estimate of how often the inequality is satisfied we decided to simulate some examples for  $x_1, x_2, x_3 \in U(0, 1)$ . We found that for 434 out of 500 samples  $\tau_3^{(12)} \geq 0$ . Of the remaining 66 out of 500 samples  $\tau_3^{(12)} < 0$ .

We will now see what happens when  $\tau_1^{(0)} = 0$ . When  $\tau_1^{(0)} = 0$ ,  $\tau_1^{(12)} = -\tau_1^{(0)} = 0$ . We have already shown that  $\tau_2^{(12)} \geq 0$  for all possible  $\tau_1^{(0)}$ ,  $\tau_2^{(0)}$  and  $\tau_3^{(0)}$ . Recall that

$$\tau_2^{(12)} = \ln \left( \frac{(1 - x_1^2 x_2^2) + \sqrt{(1 - x_1^2 x_2^2)^2 + 4x_2^2}}{2x_1 x_2} \right). \tag{A.27}$$

When  $\tau_1^{(0)} = 0$ ,  $\tau_2^{(12)}$  becomes

$$\begin{aligned}
\tau_2^{(12)} &= \ln \left( \frac{(1 - x_2^2) + \sqrt{(1 - x_2^2)^2 + 4x_2^2}}{2x_2} \right) \\
&= \ln \left( \frac{(1 - x_2^2) + \sqrt{1 - 2x_2^2 + x_2^4 + 4x_2^2}}{2x_2} \right) \\
&= \ln \left( \frac{(1 - x_2^2) + \sqrt{1 + 2x_2^2 + x_2^4}}{2x_2} \right) \\
&= \ln \left( \frac{(1 - x_2^2) + \sqrt{(1 + x_2^2)^2}}{2x_2} \right) \\
&= \ln \left( \frac{(1 - x_2^2) + \sqrt{1 + x_2^2}}{2x_2} \right) \\
&= \ln \left( \frac{2}{2x_2} \right) \\
&= \ln \left( \frac{1}{x_2} \right) \\
&= \ln \left( \frac{1}{e^{-\tau_2^{(0)}}} \right) \\
&= \ln \left( e^{\tau_2^{(0)}} \right) \\
&= \tau_2^{(0)}.
\end{aligned} \tag{A.28}$$

We will now see what happens to  $\tau_3^{(12)}$ . Recall that

$$\tau_3^{(12)} = \ln \left( \frac{-(1 - x_1^2 x_2^2) + \sqrt{(1 - x_1^2 x_2^2)^2 + 4x_2^2}}{2x_2^2 x_3} \right). \tag{A.29}$$

When  $\tau_1^{(0)} = 0$ ,  $\tau_3^{(12)}$  becomes

$$\begin{aligned}
\tau_3^{(12)} &= \ln \left( \frac{-(1-x_2^2) + \sqrt{(1-x_2^2)^2 + 4x_2^2}}{2x_2^2x_3} \right) \\
&= \ln \left( \frac{-(1-x_2^2) + \sqrt{1-2x_2^2+x_2^4+4x_2^2}}{2x_2^2x_3} \right) \\
&= \ln \left( \frac{-(1-x_2^2) + \sqrt{1+2x_2^2+x_2^4}}{2x_2^2x_3} \right) \\
&= \ln \left( \frac{-(1-x_2^2) + \sqrt{(1+x_2^2)^2}}{2x_2^2x_3} \right) \\
&= \ln \left( \frac{-(1-x_2^2) + (1+x_2^2)}{2x_2^2x_3} \right) \\
&= \ln \left( \frac{2x_2^2}{2x_2^2x_3} \right) \\
&= \ln \left( \frac{1}{x_3} \right) \\
&= \ln \left( \frac{1}{e^{-\tau_3^{(0)}}} \right) \\
&= \ln \left( e^{\tau_3^{(0)}} \right) \\
&= \tau_3^{(0)} \\
&\geq 0.
\end{aligned} \tag{A.30}$$



## Appendix B

# Time Parameters for the (13) Taxon Label Permutation

Equating the three pairwise distances for the two taxon label permutations for the three-taxon convergence-divergence network, the (13) taxon label permutation and no taxon label permutation,

$$\left\{ \begin{array}{lcl} e^{-2(\tau_1^{(0)} + \tau_2^{(0)} + \tau_3^{(0)})} & = & e^{-2(\tau_1^{(13)} + \tau_2^{(13)} + \tau_3^{(13)})}, \\ e^{-2(\tau_2^{(0)} + \tau_3^{(0)})} & = & 1 - e^{-\tau_3^{(13)}} \left( 1 - e^{-2(\tau_1^{(13)} + \tau_2^{(13)})} \right), \\ 1 - e^{-\tau_3^{(0)}} \left( 1 - e^{-2(\tau_1^{(0)} + \tau_2^{(0)})} \right) & = & e^{-2(\tau_2^{(13)} + \tau_3^{(13)})}. \end{array} \right. \quad (\text{B.1})$$

Making the appropriate substitutions as before,

$$\left\{ \begin{array}{lcl} x_1^2 x_2^2 x_3^2 & = & y_1^2 y_2^2 y_3^2, \\ x_2^2 x_3^2 & = & 1 - y_3 \left( 1 - y_1^2 y_2^2 \right), \\ 1 - x_3 \left( 1 - x_1^2 x_2^2 \right) & = & y_2^2 y_3^2. \end{array} \right. \quad (\text{B.2})$$

Making the further substitutions,

$$\left\{ \begin{array}{lcl} v_1 v_2 v_3 & = & w_1 w_2 w_3, \\ v_2 v_3 & = & 1 - w_3 + w_1 w_2, \\ 1 - v_3 + v_1 v_2 & = & w_2 w_3. \end{array} \right. \quad (\text{B.3})$$

Again, making further substitutions,

$$\begin{cases} \alpha &= w_1 w_2 w_3, \\ \beta &= 1 - w_3 + w_1 w_2, \\ \gamma &= w_2 w_3. \end{cases} \quad (\text{B.4})$$

We will now solve the system for the variables  $w_1$ ,  $w_2$  and  $w_3$ , in terms of the variables  $\alpha$ ,  $\beta$  and  $\gamma$ . We can immediately see that dividing the first equation by the third equation gives

$$w_1 = \frac{\alpha}{\gamma}. \quad (\text{B.5})$$

Now rearranging the second equation to solve for  $w_2$ ,

$$\begin{aligned} w_1 w_2 &= w_3 - (1 - \beta) \\ w_2 &= \frac{w_3 - (1 - \beta)}{w_1}. \end{aligned} \quad (\text{B.6})$$

Substituting in  $w_1 = \frac{\alpha}{\gamma}$ ,

$$\begin{aligned} w_2 &= (w_3 - (1 - \beta)) \cdot \frac{\gamma}{\alpha} \\ &= \frac{\gamma (w_3 - (1 - \beta))}{\alpha}. \end{aligned} \quad (\text{B.7})$$

Now rearranging the third equation to solve for  $w_2$ ,

$$w_2 = \frac{\gamma}{w_3}. \quad (\text{B.8})$$

Equating the two expressions for  $w_2$ ,

$$\begin{aligned} \frac{\gamma (w_3 - (1 - \beta))}{\alpha} &= \frac{\gamma}{w_3} \\ w_3 (w_3 - (1 - \beta)) &= \alpha \\ w_3^2 - (1 - \beta) w_3 - \alpha &= 0 \\ w_3 &= \frac{1}{2} \left( (1 - \beta) \pm \sqrt{(1 - \beta)^2 + 4\alpha} \right). \end{aligned} \quad (\text{B.9})$$

Demanding  $w_3 = y_3 = e^{-\tau_3^{(13)}} > 0$ , the only possible solution is

$$w_3 = \frac{1}{2} \left( (1 - \beta) + \sqrt{(1 - \beta)^2 + 4\alpha} \right). \quad (\text{B.10})$$



We can now substitute in the expression for  $w_3$  to find the expression for  $w_2$ ,

$$\begin{aligned}
w_2 &= \frac{\gamma}{w_3} \\
&= \frac{\gamma}{\frac{1}{2} \left( (1-\beta) + \sqrt{(1-\beta)^2 + 4\alpha} \right)} \\
&= \frac{2\gamma}{(1-\beta) + \sqrt{(1-\beta)^2 + 4\alpha}} \\
&= \frac{2\gamma}{(1-\beta) + \sqrt{(1-\beta)^2 + 4\alpha}} \cdot \frac{-(1-\beta) + \sqrt{(1-\beta)^2 + 4\alpha}}{-(1-\beta) + \sqrt{(1-\beta)^2 + 4\alpha}} \\
&= \frac{2\gamma \left( -(1-\beta) + \sqrt{(1-\beta)^2 + 4\alpha} \right)}{4\alpha} \\
&= \frac{\gamma}{2\alpha} \left( -(1-\beta) + \sqrt{(1-\beta)^2 + 4\alpha} \right).
\end{aligned} \tag{B.11}$$

We now have the expressions for  $w_1$ ,  $w_2$  and  $w_3$ . Recall that

$$w_1 = y_1^2 = e^{-2\tau_1^{(13)}} \tag{B.12}$$

and

$$w_1 = \frac{\alpha}{\gamma}. \tag{B.13}$$

We can now use the appropriate back-substitutions to get expressions for the time parameters on the convergence-divergence network with the (23) taxon label permutation in terms of the time parameters on the convergence-divergence network with no taxon label permutation.

The first time parameter will be

$$\begin{aligned}
\tau_1^{(13)} &= -\frac{1}{2} \ln \left( \frac{\alpha}{\gamma} \right) \\
&= \ln \sqrt{\frac{\gamma}{\alpha}} \\
&= \ln \sqrt{\frac{1 - v_3 + v_1 v_2}{v_1 v_2 v_3}} \\
&= \ln \sqrt{\frac{1 - x_3 (1 - x_1^2 x_2^2)}{x_1^2 x_2^2 x_3^2}} \\
&= \ln \sqrt{\frac{1 - e^{-\tau_3^0} \left( 1 - e^{-2(\tau_1^0 + \tau_2^0)} \right)}{e^{-2(\tau_1^0 + \tau_2^0 + \tau_3^0)}}}.
\end{aligned} \tag{B.14}$$

Recall that

$$y_2^2 = e^{-2\tau_2^{(13)}}. \quad (\text{B.15})$$

Recall that

$$w_2 = \frac{\gamma}{w_3} = y_2^2 y_3 \quad (\text{B.16})$$

and

$$w_3 = y_3. \quad (\text{B.17})$$

Consequently,

$$\begin{aligned} y_2^2 &= \frac{w_2}{w_3} \\ &= \frac{\gamma}{w_3^2} \\ &= \frac{4\gamma}{\left((1-\beta) + \sqrt{(1-\beta)^2 + 4\alpha}\right)^2}. \end{aligned} \quad (\text{B.18})$$

Again, using the appropriate substitutions,

$$\begin{aligned} \tau_2^{(13)} &= -\frac{1}{2} \ln \left( \frac{4\gamma}{\left((1-\beta) + \sqrt{(1-\beta)^2 + 4\alpha}\right)^2} \right) \\ &= \ln \left( \frac{(1-\beta) + \sqrt{(1-\beta)^2 + 4\alpha}}{2\sqrt{\gamma}} \right) \\ &= \ln \left( \frac{(1-v_2v_3) + \sqrt{(1-v_2v_3)^2 + 4v_1v_2v_3}}{2\sqrt{1-v_3+v_1v_2}} \right) \\ &= \ln \left( \frac{(1-x_2^2x_3^2) + \sqrt{(1-x_2^2x_3^2)^2 + 4x_1^2x_2^2x_3^2}}{2\sqrt{1-x_3(1-x_1^2x_2^2)}} \right) \\ &= \ln \left( \frac{\left(1 - e^{-2(\tau_2^{(0)} + \tau_3^{(0)})}\right) + \sqrt{\left(1 - e^{-2(\tau_2^{(0)} + \tau_3^{(0)})}\right)^2 + 4e^{-2(\tau_1^{(0)} + \tau_2^{(0)} + \tau_3^{(0)})}}}{2\sqrt{1 - e^{-\tau_3^{(0)}}} \left(1 - e^{-2(\tau_1^{(0)} + \tau_2^{(0)})}\right)} \right). \end{aligned} \quad (\text{B.19})$$

Recall that

$$w_3 = y_3 = e^{-\tau_3^{(13)}} \quad (\text{B.20})$$

and

$$w_3 = \frac{1}{2} \left( (1 - \beta) + \sqrt{(1 - \beta)^2 + 4\alpha} \right). \quad (\text{B.21})$$

Again, using the appropriate substitutions,

$$\begin{aligned} \tau_3^{(13)} &= -\ln \left( \frac{1}{2} \left( (1 - \beta) + \sqrt{(1 - \beta)^2 + 4\alpha} \right) \right) \\ &= \ln \left( \frac{2}{(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\alpha}} \right) \\ &= \ln \left( \frac{2}{(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\alpha}} \cdot \frac{-(1 - \beta) - \sqrt{(1 - \beta)^2 + 4\alpha}}{-(1 - \beta) - \sqrt{(1 - \beta)^2 + 4\alpha}} \right) \\ &= \ln \left( \frac{2 \left( -(1 - \beta) - \sqrt{(1 - \beta)^2 + 4\alpha} \right)}{4\alpha} \right) \\ &= \ln \left( \frac{-(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\alpha}}{2\alpha} \right) \quad (\text{B.22}) \\ &= \ln \left( \frac{-(1 - v_2 v_3) + \sqrt{(1 - v_2 v_3)^2 + 4v_1 v_2 v_3}}{2v_1 v_2 v_3} \right) \\ &= \ln \left( \frac{-(1 - x_2^2 x_3^2) + \sqrt{(1 - x_2^2 x_3^2)^2 + 4x_1^2 x_2^2 x_3^2}}{2x_1^2 x_2^2 x_3^2} \right) \\ &= \ln \left( \frac{-\left(1 - e^{-2(\tau_2^0 + \tau_3^0)}\right) + \sqrt{\left(1 - e^{-2(\tau_2^0 + \tau_3^0)}\right)^2 + 4e^{-2(\tau_1^0 + \tau_2^0 + \tau_3^0)}}}{2e^{-2(\tau_1^0 + \tau_2^0 + \tau_3^0)}} \right). \end{aligned}$$

In summary, the time parameters for the (13) permutation will be

$$\begin{cases} \tau_1^{(13)} = \ln \sqrt{\frac{1 - e^{-\tau_3^{(0)}} \left(1 - e^{-2(\tau_1^{(0)} + \tau_2^{(0)})}\right)}{e^{-2(\tau_1^{(0)} + \tau_2^{(0)} + \tau_3^{(0)})}}}, \\ \tau_2^{(13)} = \ln \left( \frac{\left(1 - e^{-2(\tau_2^{(0)} + \tau_3^{(0)})}\right) + \sqrt{\left(1 - e^{-2(\tau_2^{(0)} + \tau_3^{(0)})}\right)^2 + 4e^{-2(\tau_1^{(0)} + \tau_2^{(0)} + \tau_3^{(0)})}}}{2\sqrt{1 - e^{-\tau_3^{(0)}} \left(1 - e^{-2(\tau_1^{(0)} + \tau_2^{(0)})}\right)}} \right), \\ \tau_3^{(13)} = \ln \left( \frac{-\left(1 - e^{-2(\tau_2^{(0)} + \tau_3^{(0)})}\right) + \sqrt{\left(1 - e^{-2(\tau_2^{(0)} + \tau_3^{(0)})}\right)^2 + 4e^{-2(\tau_1^{(0)} + \tau_2^{(0)} + \tau_3^{(0)})}}}{2e^{-2(\tau_1^{(0)} + \tau_2^{(0)} + \tau_3^{(0)})}} \right). \end{cases} \quad (\text{B.23})$$

Recall that  $\tau_1^{(13)}$  can be expressed as

$$\tau_1^{(13)} = \ln \sqrt{\frac{1 - x_3 (1 - x_1^2 x_2^2)}{x_1^2 x_2^2 x_3^2}}. \quad (\text{B.24})$$

Demanding  $0 < x_1, x_2, x_3 \leq 1$  and  $\tau_1^{(13)} \geq 0$ ,

$$\begin{aligned} 1 - x_3 (1 - x_1^2 x_2^2) &\geq x_1^2 x_2^2 x_3^2 \\ \Leftrightarrow 1 - x_3 (1 - x_1^2 x_2^2) - x_1^2 x_2^2 x_3^2 &\geq 0 \\ \Leftrightarrow 1 - x_3 (1 - x_1^2 x_2^2 + x_1^2 x_2^2 x_3) &\geq 0 \\ \Leftrightarrow 1 - x_3 (1 - x_1^2 x_2^2 (1 - x_3)) &\geq 0. \end{aligned} \quad (\text{B.25})$$

Since  $0 < x_1, x_2, x_3 \leq 1$ ,

$$\begin{aligned} 0 &< x_3 \leq 1 \\ \Leftrightarrow -1 &\leq -x_3 < 0 \\ \Leftrightarrow 0 &\leq 1 - x_3 < 1 \\ \Leftrightarrow 0 &\leq x_1^2 x_2^2 (1 - x_3) < 1 \\ \Leftrightarrow -1 &< -x_1^2 x_2^2 (1 - x_3) \leq 0 \\ \Leftrightarrow 0 &< 1 - x_1^2 x_2^2 (1 - x_3) \leq 1 \\ \Leftrightarrow 0 &< x_3 (1 - x_1^2 x_2^2 (1 - x_3)) \leq 1 \\ \Leftrightarrow -1 &\leq -x_3 (1 - x_1^2 x_2^2 (1 - x_3)) < 0 \\ \Leftrightarrow 0 &\leq 1 - x_3 (1 - x_1^2 x_2^2 (1 - x_3)) < 1. \end{aligned} \quad (\text{B.26})$$

We can conclude that  $\tau_1^{(13)} \geq 0$ .

Recall that  $\tau_2^{(13)}$  can be expressed as

$$\begin{aligned}
\tau_2^{(13)} &= \ln \left( \frac{(1-\beta) + \sqrt{(1-\beta)^2 + 4\alpha}}{2\sqrt{\gamma}} \right) \\
&= \ln \left( \frac{(1-\beta) + \sqrt{(1-\beta)^2 + 4\alpha}}{2\sqrt{\gamma}} \cdot \frac{-(1-\beta) + \sqrt{(1-\beta)^2 + 4\alpha}}{-(1-\beta) + \sqrt{(1-\beta)^2 + 4\alpha}} \right) \\
&= \ln \left( \frac{4\alpha}{2\sqrt{\gamma} \left( -(1-\beta) + \sqrt{(1-\beta)^2 + 4\alpha} \right)} \right) \\
&= \ln \left( \frac{2\alpha}{\sqrt{\gamma} \left( -(1-\beta) + \sqrt{(1-\beta)^2 + 4\alpha} \right)} \right).
\end{aligned} \tag{B.27}$$

Demanding  $0 < \alpha, \beta, \gamma \leq 1$  and  $\tau_2^{(13)} \geq 0$ ,

$$\begin{aligned}
2\alpha &\geq \sqrt{\gamma} \left( -(1-\beta) + \sqrt{(1-\beta)^2 + 4\alpha} \right) \\
&\Leftrightarrow \frac{2\alpha}{\sqrt{\gamma}} \geq -(1-\beta) + \sqrt{(1-\beta)^2 + 4\alpha} \\
&\Leftrightarrow \frac{2\alpha}{\sqrt{\gamma}} + (1-\beta) \geq \sqrt{(1-\beta)^2 + 4\alpha} \\
&\Leftrightarrow \frac{4\alpha^2}{\gamma} + (1-\beta)^2 + \frac{4\alpha(1-\beta)}{\sqrt{\gamma}} \geq (1-\beta)^2 + 4\alpha \\
&\Leftrightarrow \frac{4\alpha^2}{\gamma} + \frac{4\alpha(1-\beta)}{\sqrt{\gamma}} \geq 4\alpha \\
&\Leftrightarrow \frac{\alpha}{\gamma} + \frac{(1-\beta)}{\sqrt{\gamma}} \geq 1 \\
&\Leftrightarrow \alpha + (1-\beta)\sqrt{\gamma} \geq \gamma \\
&\Leftrightarrow (1-\beta)\sqrt{\gamma} \geq \gamma - \alpha.
\end{aligned} \tag{B.28}$$

Substituting in the expressions for  $\alpha, \beta$  and  $\gamma$  in terms of  $x_1, x_2$  and  $x_3$ ,

$$\begin{aligned}
(1-x_2^2x_3^2) \sqrt{1-x_3(1-x_1^2x_2^2)} &\geq 1-x_3(1-x_1^2x_2^2) - x_1^2x_2^2x_3^2 \\
&\Leftrightarrow (1-x_2^2x_3^2) \sqrt{1-x_3(1-x_1^2x_2^2)} \geq 1-x_3+x_1^2x_2^2x_3-x_1^2x_2^2x_3^2 \\
&\Leftrightarrow (1-x_2^2x_3^2) \sqrt{1-x_3(1-x_1^2x_2^2)} \geq 1-x_3(1-x_1^2x_2^2+x_1^2x_2^2x_3) \\
&\Leftrightarrow (1-x_2^2x_3^2) \sqrt{1-x_3(1-x_1^2x_2^2)} \geq 1-x_3(1-x_1^2x_2^2(1-x_3)).
\end{aligned} \tag{B.29}$$

Since  $0 < x_2, x_3 \leq 1$ ,

$$\begin{aligned}
& 0 < x_2^2 x_3^2 \leq 1 \\
& \Leftrightarrow -1 \leq -x_2^2 x_3^2 < 0 \\
& \Leftrightarrow 0 \leq 1 - x_2^2 x_3^2 < 1.
\end{aligned} \tag{B.30}$$

Similarly, since  $0 < x_1, x_2, x_3 \leq 1$ ,

$$\begin{aligned}
& 0 < x_1^2 x_2^2 \leq 1 \\
& \Leftrightarrow -1 \leq -x_1^2 x_2^2 < 0 \\
& \Leftrightarrow 0 \leq 1 - x_1^2 x_2^2 < 1 \\
& \Leftrightarrow 0 \leq x_3 (1 - x_1^2 x_2^2) < 1 \\
& \Leftrightarrow -1 < -x_3 (1 - x_1^2 x_2^2) \leq 0 \\
& \Leftrightarrow 0 < 1 - x_3 (1 - x_1^2 x_2^2) \leq 1 \\
& \Leftrightarrow 0 < \sqrt{1 - x_3 (1 - x_1^2 x_2^2)} \leq 1.
\end{aligned} \tag{B.31}$$

We can conclude that the left hand side of the inequality must be in the minimum length interval,

$$0 \leq (1 - x_2^2 x_3^2) \sqrt{1 - x_3 (1 - x_1^2 x_2^2)} < 1. \tag{B.32}$$

Similarly, recall from (B.26) on page 190 that the minimum length interval for the right hand side of the equation will be

$$0 \leq 1 - x_3 (1 - x_1^2 x_2^2 (1 - x_3)) < 1. \tag{B.33}$$

It is not immediately clear how  $\tau_2^{(13)}$  is constrained. As a simple check we will substitute in some values for  $x_1$ ,  $x_2$  and  $x_3$ . First, suppose  $x_3 = 1$ . The inequality for  $\tau_2^{(13)} \geq 0$  then becomes

$$\begin{aligned}
(1 - x_2^2) \sqrt{1 - (1 - x_1^2 x_2^2)} &\geq 1 - (1 - x_1^2 x_2^2 (1 - x_3)) \\
(1 - x_2^2) \sqrt{x_1^2 x_2^2} &\geq 0 \\
x_1 x_2 (1 - x_2^2) &\geq 0.
\end{aligned} \tag{B.34}$$

Clearly this will be true for all  $0 < x_1, x_2 \leq 1$ . We can conclude that  $\tau_2^{(13)}$  can at least sometimes be non-negative. As a second example, suppose  $\epsilon$  is a “small” positive number and  $x_1 = \epsilon$ ,  $x_2 = 1$

and  $x_3 = 1 - \epsilon$ . The inequality for  $\tau_2^{(13)} \geq 0$  then becomes

$$\begin{aligned}
& \left(1 - (1 - \epsilon)^2\right) \sqrt{1 - (1 - \epsilon)(1 - \epsilon^2)} \geq 1 - (1 - \epsilon)(1 - \epsilon^2(1 - (1 - \epsilon))) \\
\Leftrightarrow & \left(1 - (1 - 2\epsilon + \epsilon^2)\right) \sqrt{1 - (1 - \epsilon - \epsilon^2 + \epsilon^3)} \geq 1 - (1 - \epsilon)(1 - \epsilon^3) \\
& \Leftrightarrow (2\epsilon - \epsilon^2) \sqrt{\epsilon + \epsilon^2 - \epsilon^3} \geq 1 - (1 - \epsilon - \epsilon^3 + \epsilon^4) \\
& \Leftrightarrow \epsilon(2 - \epsilon) \sqrt{\epsilon(1 + \epsilon - \epsilon^2)} \geq \epsilon + \epsilon^3 - \epsilon^4 \\
& \Leftrightarrow \sqrt{\epsilon}(2 - \epsilon) \sqrt{1 + \epsilon - \epsilon^2} \geq 1 + \epsilon^2 - \epsilon^3 \\
& \Leftrightarrow \sqrt{\epsilon}(2 - \epsilon) \sqrt{1 + \epsilon(1 - \epsilon)} \geq 1 + \epsilon^2(1 - \epsilon).
\end{aligned} \tag{B.35}$$

The left hand side of the inequality must be some “small” positive number, while the right hand side of the inequality is greater than 1. We can conclude that the inequality is not always satisfied and  $\tau_2^{(13)}$  can sometimes be non-negative and sometimes be negative.

Again, simulating some examples for  $x_1, x_2, x_3 \in U(0, 1)$  we found that for 497 out of 500 samples  $\tau_3^{(13)} \geq 0$ . Of the remaining 3 out of 500 samples  $\tau_3^{(13)} < 0$ .

Recall that  $\tau_3^{(13)}$  can be expressed as

$$\tau_3^{(13)} = \ln \left( \frac{-(1 - x_2^2 x_3^2) + \sqrt{(1 - x_2^2 x_3^2)^2 + 4x_1^2 x_2^2 x_3^2}}{2x_1^2 x_2^2 x_3^2} \right). \tag{B.36}$$

Demanding  $0 < x_1, x_2, x_3 \leq 1$  and  $\tau_3^{(13)} \geq 0$ ,

$$\begin{aligned}
& -(1 - x_2^2 x_3^2) + \sqrt{(1 - x_2^2 x_3^2)^2 + 4x_1^2 x_2^2 x_3^2} \geq 2x_1^2 x_2^2 x_3^2 \\
& \Leftrightarrow \sqrt{(1 - x_2^2 x_3^2)^2 + 4x_1^2 x_2^2 x_3^2} \geq (1 - x_2^2 x_3^2) + 2x_1^2 x_2^2 x_3^2 \\
& \Leftrightarrow (1 - x_2^2 x_3^2)^2 + 4x_1^2 x_2^2 x_3^2 \geq (1 - x_2^2 x_3^2)^2 + 4x_1^4 x_2^4 x_3^4 + 4x_1^2 x_2^2 x_3^2 (1 - x_2^2 x_3^2) \\
& \Leftrightarrow 4x_1^2 x_2^2 x_3^2 \geq 4x_1^4 x_2^4 x_3^4 + 4x_1^2 x_2^2 x_3^2 (1 - x_2^2 x_3^2) \\
& \Leftrightarrow 1 \geq x_1^2 x_2^2 x_3^2 + (1 - x_2^2 x_3^2) \\
& \Leftrightarrow 0 \geq x_1^2 x_2^2 x_3^2 - x_2^2 x_3^2 \\
& \Leftrightarrow x_2^2 x_3^2 \geq x_1^2 x_2^2 x_3^2 \\
& \Leftrightarrow 1 \geq x_1^2 \\
& \Leftrightarrow 1 \geq x_1.
\end{aligned} \tag{B.37}$$

We can conclude that  $\tau_3^{(13)} \geq 0$ .





## Appendix C

# Time Parameters for the (23) Taxon Label Permutation

Equating the three pairwise distances for the two taxon label permutations for the three-taxon convergence-divergence network, the (23) taxon label permutation and no taxon label permutation,

$$\left\{ \begin{array}{l} 2 \left( \tau_2^{(0)} + \tau_3^{(0)} \right) = 2 \left( \tau_2^{(23)} + \tau_3^{(23)} \right), \\ 2 \left( \tau_1^{(0)} + \tau_2^{(0)} + \tau_3^{(0)} \right) = -\ln \left( 1 - e^{-\tau_3^{(23)}} \left( 1 - e^{-2 \left( \tau_1^{(23)} + \tau_2^{(23)} \right)} \right) \right), \\ -\ln \left( 1 - e^{-\tau_3^{(0)}} \left( 1 - e^{-2 \left( \tau_1^{(0)} + \tau_2^{(0)} \right)} \right) \right) = 2 \left( \tau_1^{(23)} + \tau_2^{(23)} + \tau_3^{(23)} \right). \end{array} \right. \quad (\text{C.1})$$

Making the appropriate substitutions as before,

$$\left\{ \begin{array}{l} x_2^2 x_3^2 = y_2^2 y_3^2, \\ x_1^2 x_2^2 x_3^2 = 1 - y_3 \left( 1 - y_1^2 y_2^2 \right), \\ 1 - x_3 \left( 1 - x_1^2 x_2^2 \right) = y_1^2 y_2^2 y_3^2. \end{array} \right. \quad (\text{C.2})$$

Making the further substitutions,

$$\left\{ \begin{array}{l} v_2 v_3 = w_2 w_3, \\ v_1 v_2 v_3 = 1 - w_3 + w_1 w_2, \\ 1 - v_3 + v_1 v_2 = w_1 w_2 w_3. \end{array} \right. \quad (\text{C.3})$$

Again, making the further substitutions,

$$\begin{cases} \alpha &= w_2 w_3, \\ \beta &= 1 - w_3 + w_1 w_2, \\ \gamma &= w_1 w_2 w_3. \end{cases} \quad (\text{C.4})$$

We will now solve the system for the variables,  $w_1$ ,  $w_2$  and  $w_3$ , in terms of the variables,  $\alpha$ ,  $\beta$  and  $\gamma$ . We can immediately see that dividing the third equation by the first equation gives

$$w_1 = \frac{\gamma}{\alpha}. \quad (\text{C.5})$$

Now rearranging the first equation to solve for  $w_2$ ,

$$w_2 = \frac{\alpha}{w_3}. \quad (\text{C.6})$$

Substituting in the expressions for  $w_1$  and  $w_2$  into the second equation,

$$\begin{aligned} \beta &= 1 - w_3 + \frac{\gamma}{\alpha} \frac{\alpha}{w_3} \\ \Leftrightarrow \beta &= 1 - w_3 + \frac{\gamma}{w_3} \\ \Leftrightarrow -(1 - \beta) &= -w_3 + \frac{\gamma}{w_3} \\ \Leftrightarrow w_3 - (1 - \beta) - \frac{\gamma}{w_3} &= 0 \\ \Leftrightarrow w_3^2 - (1 - \beta) w_3 - \gamma &= 0 \\ \Leftrightarrow w_3 &= \frac{1}{2} \left( (1 - \beta) \pm \sqrt{(1 - \beta)^2 + 4\gamma} \right). \end{aligned} \quad (\text{C.7})$$

Demanding  $w_3 = y_3 = e^{-\tau_3^{(23)}} > 0$ , the only possible solution is

$$w_3 = \frac{1}{2} \left( (1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma} \right). \quad (\text{C.8})$$

We can now substitute in the expression for  $w_3$  to find the expression for  $w_2$ ,

$$\begin{aligned}
w_2 &= \frac{\alpha}{w_3} \\
&= \frac{\alpha}{\frac{1}{2} \left( (1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma} \right)} \\
&= \frac{2\alpha}{(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma}} \\
&= \frac{2\alpha}{(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma}} \cdot \frac{-(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma}}{-(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma}} \\
&= \frac{2\alpha \left( -(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma} \right)}{4\gamma} \\
&= \frac{\alpha}{2\gamma} \left( -(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma} \right).
\end{aligned} \tag{C.9}$$

We now have the expressions for  $w_1$ ,  $w_2$  and  $w_3$ . Recall that

$$w_1 = y_1^2 = e^{-2\tau_1^{(23)}} \tag{C.10}$$

and

$$w_1 = \frac{\gamma}{\alpha}. \tag{C.11}$$

We can now use the appropriate back-substitutions to get expressions for the time parameters on the convergence-divergence network with the (23) taxon label permutation in terms of the time parameters on the convergence-divergence network with no taxon label permutation.

The first time parameter will be

$$\begin{aligned}
\tau_1^{(23)} &= -\frac{1}{2} \ln \left( \frac{\gamma}{\alpha} \right) \\
&= \ln \sqrt{\frac{\alpha}{\gamma}} \\
&= \ln \sqrt{\frac{v_2 v_3}{1 - v_3 + v_1 v_2}} \\
&= \ln \sqrt{\frac{x_2^2 x_3^2}{1 - x_3 (1 - x_1^2 x_2^2)}} \\
&= \ln \sqrt{\frac{e^{-2(\tau_2^{(0)} + \tau_3^{(0)})}}{1 - e^{-\tau_3^{(0)}} \left( 1 - e^{-2(\tau_1^{(0)} + \tau_2^{(0)})} \right)}}.
\end{aligned} \tag{C.12}$$

Recall that

$$y_2^2 = e^{-2\tau_2^{(23)}}. \quad (\text{C.13})$$

Recall that

$$w_2 = \frac{\alpha}{w_3} = y_2^2 y_3 \quad (\text{C.14})$$

and

$$w_3 = y_3. \quad (\text{C.15})$$

Consequently,

$$\begin{aligned} y_2^2 &= \frac{w_2}{w_3} \\ &= \frac{\alpha}{w_3^2} \\ &= \frac{4\alpha}{\left((1-\beta) + \sqrt{(1-\beta)^2 + 4\gamma}\right)^2}. \end{aligned} \quad (\text{C.16})$$

Again, using the appropriate substitutions,

$$\begin{aligned} \tau_2^{(23)} &= -\frac{1}{2} \ln \left( \frac{4\alpha}{\left((1-\beta) + \sqrt{(1-\beta)^2 + 4\gamma}\right)^2} \right) \\ &= \ln \left( \frac{(1-\beta) + \sqrt{(1-\beta)^2 + 4\gamma}}{2\sqrt{\alpha}} \right) \\ &= \ln \left( \frac{(1-v_1v_2v_3) + \sqrt{(1-v_1v_2v_3)^2 + 4(1-v_3+v_1v_2)}}{2\sqrt{v_2v_3}} \right) \\ &= \ln \left( \frac{(1-x_1^2x_2^2x_3^2) + \sqrt{(1-x_1^2x_2^2x_3^2)^2 + 4(1-x_3(1-x_1^2x_2^2))}}{2\sqrt{x_2^2x_3^2}} \right) \\ &= \ln \left( \frac{(1-x_1^2x_2^2x_3^2) + \sqrt{(1-x_1^2x_2^2x_3^2)^2 + 4(1-x_3(1-x_1^2x_2^2))}}{2x_2x_3} \right) \\ &= \ln \left( \frac{\left(1-e^{-2(\tau_1^{(0)}+\tau_2^{(0)}+\tau_3^{(0)})}\right) + \sqrt{\left(1-e^{-2(\tau_1^{(0)}+\tau_2^{(0)}+\tau_3^{(0)})}\right)^2 + 4\left(1-e^{-\tau_3^{(0)}}\left(1-e^{-2(\tau_1^{(0)}+\tau_2^{(0)})}\right)\right)}}{2e^{-(\tau_2^{(0)}+\tau_3^{(0)})}} \right). \end{aligned} \quad (\text{C.17})$$

Recall that

$$w_3 = y_3 = e^{-\tau_3^{(23)}} \quad (\text{C.18})$$

and

$$w_3 = \frac{1}{2} \left( (1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma} \right). \quad (\text{C.19})$$

Again, using the appropriate substitutions,

$$\begin{aligned} \tau_3^{(23)} &= -\ln \frac{1}{2} \left( (1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma} \right) \\ &= \ln \left( \frac{2}{(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma}} \right) \\ &= \ln \left( \frac{2}{(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma}} \cdot \frac{-(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma}}{-(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma}} \right) \\ &= \ln \left( \frac{-2(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma}}{4\gamma} \right) \\ &= \ln \left( \frac{-(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma}}{2\gamma} \right) \quad (\text{C.20}) \\ &= \ln \left( \frac{-(1 - v_1 v_2 v_3) + \sqrt{(1 - v_1 v_2 v_3)^2 + 4(1 - v_3 + v_1 v_2)}}{2(1 - v_3 + v_1 v_2)} \right) \\ &= \ln \left( \frac{-(1 - x_1^2 x_2^2 x_3^2) + \sqrt{(1 - x_1^2 x_2^2 x_3^2)^2 + 4(1 - x_3(1 - x_1^2 x_2^2))}}{2(1 - x_3(1 - x_1^2 x_2^2))} \right) \\ &= \ln \left( \frac{-\left(1 - e^{-2(\tau_1^0 + \tau_2^0 + \tau_3^0)}\right) + \sqrt{\left(1 - e^{-2(\tau_1^0 + \tau_2^0 + \tau_3^0)}\right)^2 + 4\left(1 - e^{-\tau_3^0} \left(1 - e^{-2(\tau_1^0 + \tau_2^0)}\right)\right)}}{2\left(1 - e^{-\tau_3^0} \left(1 - e^{-2(\tau_1^0 + \tau_2^0)}\right)\right)} \right). \end{aligned}$$

In summary,

$$\begin{cases} \tau_1^{(23)} &= \ln \sqrt{\frac{e^{-2(\tau_2^0 + \tau_3^0)}}{1 - e^{-\tau_3^0} \left(1 - e^{-2(\tau_1^0 + \tau_2^0)}\right)}}, \\ \tau_2^{(23)} &= \ln \left( \frac{\left(1 - e^{-2(\tau_1^0 + \tau_2^0 + \tau_3^0)}\right) + \sqrt{\left(1 - e^{-2(\tau_1^0 + \tau_2^0 + \tau_3^0)}\right)^2 + 4 \left(1 - e^{-\tau_3^0} \left(1 - e^{-2(\tau_1^0 + \tau_2^0)}\right)\right)}}{2e^{-(\tau_2^0 + \tau_3^0)}} \right), \\ \tau_3^{(23)} &= \ln \left( \frac{-\left(1 - e^{-2(\tau_1^0 + \tau_2^0 + \tau_3^0)}\right) + \sqrt{\left(1 - e^{-2(\tau_1^0 + \tau_2^0 + \tau_3^0)}\right)^2 + 4 \left(1 - e^{-\tau_3^0} \left(1 - e^{-2(\tau_1^0 + \tau_2^0)}\right)\right)}}{2 \left(1 - e^{-\tau_3^0} \left(1 - e^{-2(\tau_1^0 + \tau_2^0)}\right)\right)} \right). \end{cases} \quad (\text{C.21})$$

Recall that  $\tau_1^{(23)}$  can be expressed as

$$\ln \sqrt{\frac{x_2^2 x_3^2}{1 - x_3 (1 - x_1^2 x_2^2)}}. \quad (\text{C.22})$$

Demanding  $0 < x_1, x_2, x_3 \leq 1$  and  $\tau_1^{(23)} \geq 0$ ,

$$\begin{aligned} x_2^2 x_3^2 &\geq 1 - x_3 (1 - x_1^2 x_2^2) \\ \Leftrightarrow x_2^2 x_3^2 &\geq 1 - x_3 + x_1^2 x_2^2 x_3 \\ \Leftrightarrow 0 &\geq 1 - x_3 + x_1^2 x_2^2 x_3 - x_2^2 x_3^2 \\ \Leftrightarrow 0 &\geq 1 - x_3 (1 - x_1^2 x_2^2 + x_2^2 x_3) \\ \Leftrightarrow 0 &\geq 1 - x_3 (1 - x_2^2 (x_1^2 - x_3)). \end{aligned} \quad (\text{C.23})$$

This is the same inequality as in (A.25) on page 181, except with the direction of the inequality swapped. We can conclude that the inequality will not be satisfied when  $x_3 < x_1^2$ . Clearly, when  $x_3 = x_1^2$  the inequality will be satisfied, since  $1 - x_3 > 0$ . We can conclude that  $\tau_1^{(23)}$  can sometimes be non-negative and sometimes be negative.

To get a rough estimate of how often the inequality is satisfied we decided to simulate some samples for  $x_1, x_2, x_3 \in U(0, 1)$ . We found that for 52 out of 500 samples  $\tau_1^{(23)} \geq 0$ . Of the remaining 448 out of 500 samples  $\tau_1^{(23)} < 0$ .

Recall that

$$\beta = x_1^2 x_2^2 x_3^2 \quad (\text{C.24})$$

and

$$\gamma = 1 - x_3 (1 - x_1^2 x_2^2). \quad (\text{C.25})$$

To determine whether  $\tau_2^{(23)} \geq 0$  we first will see whether  $\beta \leq \gamma$  is true. This is the statement,

$$\begin{aligned}
 x_1^2 x_2^2 x_3^2 &\leq 1 - x_3 (1 - x_1^2 x_2^2) \\
 &\Leftrightarrow 0 \leq 1 - x_3 + x_1^2 x_2^2 - x_1^2 x_2^2 x_3^2 \\
 &\Leftrightarrow 0 \leq 1 - x_3 (1 - x_1^2 x_2^2 + x_1^2 x_2^2 x_3) \\
 &\Leftrightarrow 0 \leq 1 - x_3 (1 - x_1^2 x_2^2 (1 - x_3)).
 \end{aligned} \tag{C.26}$$

From (B.26) on page 190 we can conclude that  $\beta \leq \gamma$ . Recall that

$$\tau_2^{(23)} = \ln \left( \frac{(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma}}{2\sqrt{\alpha}} \right). \tag{C.27}$$

Demanding  $0 < \alpha, \beta, \gamma \leq 1$  and  $\tau_2^{(23)} \geq 0$ ,

$$(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma} \geq 2\sqrt{\alpha}. \tag{C.28}$$

Since  $\beta \leq \gamma$ ,

$$\begin{aligned}
 -\beta &\geq -\gamma \\
 1 - \beta &\leq 1 - \gamma.
 \end{aligned} \tag{C.29}$$

We can conclude that

$$\begin{aligned}
 (1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma} &\geq (1 - \gamma) + \sqrt{(1 - \gamma)^2 + 4\gamma} \\
 &= (1 - \gamma) + \sqrt{1 - 2\gamma + \gamma^2 + 4\gamma} \\
 &= (1 - \gamma) + \sqrt{1 + 2\gamma + \gamma^2} \\
 &= (1 - \gamma) + \sqrt{(1 + \gamma)^2} \\
 &= (1 - \gamma) + (1 + \gamma) \\
 &= 2 \\
 &\geq 2\sqrt{\alpha}.
 \end{aligned} \tag{C.30}$$

We can conclude that  $\tau_2^{(23)} \geq 0$ .

Recall that  $\tau_3^{(23)}$  can be expressed as

$$\tau_3^{(23)} = \ln \left( \frac{-(1 - x_1^2 x_2^2 x_3^2) + \sqrt{(1 - x_1^2 x_2^2 x_3^2)^2 + 4(1 - x_3(1 - x_1^2 x_2^2))}}{2(1 - x_3(1 - x_1^2 x_2^2))} \right). \tag{C.31}$$

Demanding  $0 < x_1, x_2, x_3 \leq 1$  and  $\tau_3^{(23)} \geq 0$ ,

$$\begin{aligned}
& - (1 - x_1^2 x_2^2 x_3^2) + \sqrt{(1 - x_1^2 x_2^2 x_3^2)^2 + 4(1 - x_3(1 - x_1^2 x_2^2))} \geq 2(1 - x_3(1 - x_1^2 x_2^2)) \\
& \Leftrightarrow \sqrt{(1 - x_1^2 x_2^2 x_3^2)^2 + 4(1 - x_3(1 - x_1^2 x_2^2))} \geq (1 - x_1^2 x_2^2 x_3^2) \\
& \quad + 2(1 - x_3(1 - x_1^2 x_2^2)) \\
& \Leftrightarrow (1 - x_1^2 x_2^2 x_3^2)^2 + 4(1 - x_3(1 - x_1^2 x_2^2)) \geq (1 - x_1^2 x_2^2 x_3^2)^2 \\
& \quad + 4(1 - x_3(1 - x_1^2 x_2^2))^2 \\
& \quad + 4(1 - x_1^2 x_2^2 x_3^2)(1 - x_3(1 - x_1^2 x_2^2)) \\
& \Leftrightarrow 4(1 - x_3(1 - x_1^2 x_2^2)) \geq 4(1 - x_3(1 - x_1^2 x_2^2))^2 \\
& \quad + 4(1 - x_1^2 x_2^2 x_3^2)(1 - x_3(1 - x_1^2 x_2^2)) \\
& \Leftrightarrow 1 \geq (1 - x_3(1 - x_1^2 x_2^2)) \\
& \quad + (1 - x_1^2 x_2^2 x_3^2) \\
& \Leftrightarrow 0 \geq -x_3 + x_1^2 x_2^2 x_3 + 1 - x_1^2 x_2^2 x_3^2 \\
& \Leftrightarrow 0 \geq 1 - x_3(1 - x_1^2 x_2^2 + x_1^2 x_2^2 x_3) \\
& \Leftrightarrow 0 \geq 1 - x_3(1 - x_1^2 x_2^2(1 - x_3)).
\end{aligned} \tag{C.32}$$

This is the same inequality as in (B.26) on page 190, except with the direction of the inequality swapped. We can conclude that  $\tau_3^{(23)} \leq 0$ .



## Appendix D

# The (23) Permutation in Terms of the (12) Permutation

Equating the three pairwise distances for the two taxon label permutations for the three-taxon convergence-divergence network, the (23) taxon label permutation and the (12) taxon label permutation,

$$\left\{ \begin{array}{lcl} 2 \left( \tau_1^{(12)} + \tau_2^{(12)} + \tau_3^{(12)} \right) & = & 2 \left( \tau_2^{(23)} + \tau_3^{(23)} \right), \\ 2 \left( \tau_2^{(12)} + \tau_3^{(12)} \right) & = & -\ln \left( 1 - e^{-\tau_3^{(23)}} \left( 1 - e^{-2 \left( \tau_1^{(23)} + \tau_2^{(23)} \right)} \right) \right), \\ -\ln \left( 1 - e^{-\tau_3^{(12)}} \left( 1 - e^{-2 \left( \tau_1^{(12)} + \tau_2^{(12)} \right)} \right) \right) & = & 2 \left( \tau_1^{(23)} + \tau_2^{(23)} + \tau_3^{(23)} \right). \end{array} \right. \quad (\text{D.1})$$

Making the appropriate substitutions as before,

$$\left\{ \begin{array}{lcl} x_1^2 x_2^2 x_3^2 & = & y_2^2 y_3^2, \\ x_2^2 x_3^2 & = & 1 - y_3 \left( 1 - y_1^2 y_2^2 \right), \\ 1 - x_3 \left( 1 - x_1^2 x_2^2 \right) & = & y_1^2 y_2^2 y_3^2. \end{array} \right. \quad (\text{D.2})$$

Making the further substitutions,

$$\left\{ \begin{array}{lcl} v_1 v_2 v_3 & = & w_2 w_3, \\ v_2 v_3 & = & 1 - w_3 + w_1 w_2, \\ 1 - v_3 + v_1 v_2 & = & w_1 w_2 w_3. \end{array} \right. \quad (\text{D.3})$$

Again, making the further substitutions,

$$\begin{cases} \alpha &= w_2 w_3, \\ \beta &= 1 - w_3 + w_1 w_2, \\ \gamma &= w_1 w_2 w_3. \end{cases} \quad (\text{D.4})$$

We will now solve the system for the variables,  $w_1$ ,  $w_2$  and  $w_3$ , in terms of the variables,  $\alpha$ ,  $\beta$  and  $\gamma$ . We can immediately see that dividing the third equation by the first equation gives

$$w_1 = \frac{\gamma}{\alpha}. \quad (\text{D.5})$$

Now rearranging the first equation to solve for  $w_2$ ,

$$w_2 = \frac{\alpha}{w_3}. \quad (\text{D.6})$$

Substituting in the expressions for  $w_1$  and  $w_2$  into the second equation,

$$\begin{aligned} \beta &= 1 - w_3 + \frac{\gamma}{\alpha} \frac{\alpha}{w_3} \\ \Leftrightarrow \beta &= 1 - w_3 + \frac{\gamma}{w_3} \\ \Leftrightarrow -(1 - \beta) &= -w_3 + \frac{\gamma}{w_3} \\ \Leftrightarrow w_3 - (1 - \beta) - \frac{\gamma}{w_3} &= 0 \\ \Leftrightarrow w_3^2 - (1 - \beta) w_3 - \gamma &= 0 \\ \Leftrightarrow w_3 &= \frac{1}{2} \left( (1 - \beta) \pm \sqrt{(1 - \beta)^2 + 4\gamma} \right). \end{aligned} \quad (\text{D.7})$$

Demanding  $w_3 = y_3 = e^{-\tau_3^{(23)}} > 0$ , the only possible solution is

$$w_3 = \frac{1}{2} \left( (1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma} \right). \quad (\text{D.8})$$

We can now substitute in the expression for  $w_3$  to find the expression for  $w_2$ ,

$$\begin{aligned}
w_2 &= \frac{\alpha}{w_3} \\
&= \frac{\alpha}{\frac{1}{2} \left( (1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma} \right)} \\
&= \frac{2\alpha}{(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma}} \\
&= \frac{2\alpha}{(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma}} \cdot \frac{-(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma}}{-(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma}} \\
&= \frac{2\alpha \left( -(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma} \right)}{4\gamma} \\
&= \frac{\alpha}{2\gamma} \left( -(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma} \right).
\end{aligned} \tag{D.9}$$

We now have the expressions for  $w_1$ ,  $w_2$  and  $w_3$ . Recall that

$$w_1 = y_1^2 = e^{-2\tau_1^{(23)}} \tag{D.10}$$

and

$$w_1 = \frac{\gamma}{\alpha}. \tag{D.11}$$

We can now use the appropriate back-substitutions to get expressions for the time parameters on the convergence-divergence network with the (23) taxon label permutation in terms of the time parameters on the convergence-divergence network with the (12) taxon label permutation.

The first time parameter will be

$$\begin{aligned}
\tau_1^{(23)} &= -\frac{1}{2} \ln \left( \frac{\gamma}{\alpha} \right) \\
&= \ln \sqrt{\frac{\alpha}{\gamma}} \\
&= \ln \sqrt{\frac{v_1 v_2 v_3}{1 - v_3 + v_1 v_2}} \\
&= \ln \sqrt{\frac{x_1^2 x_2^2 x_3^2}{1 - x_3 (1 - x_1^2 x_2^2)}} \\
&= \ln \sqrt{\frac{e^{-2(\tau_1^{(12)} + \tau_2^{(12)} + \tau_3^{(12)})}}{1 - e^{-\tau_3^{(12)}} \left( 1 - e^{-2(\tau_1^{(12)} + \tau_2^{(12)})} \right)}}}
\end{aligned} \tag{D.12}$$

Recall that

$$y_2^2 = e^{-2\tau_2^{(23)}}. \quad (\text{D.13})$$

Recall that

$$w_2 = \frac{\alpha}{w_3} = y_2^2 y_3 \quad (\text{D.14})$$

and

$$w_3 = y_3. \quad (\text{D.15})$$

Consequently,

$$\begin{aligned} y_2^2 &= \frac{w_2}{w_3} \\ &= \frac{\alpha}{w_3^2} \\ &= \frac{4\alpha}{\left((1-\beta) + \sqrt{(1-\beta)^2 + 4\gamma}\right)^2}. \end{aligned} \quad (\text{D.16})$$

Again, using the appropriate substitutions,

$$\begin{aligned} \tau_2^{(23)} &= -\frac{1}{2} \ln \left( \frac{4\alpha}{\left((1-\beta) + \sqrt{(1-\beta)^2 + 4\gamma}\right)^2} \right) \\ &= \ln \left( \frac{(1-\beta) + \sqrt{(1-\beta)^2 + 4\gamma}}{2\sqrt{\alpha}} \right) \\ &= \ln \left( \frac{(1-v_2v_3) + \sqrt{(1-v_2v_3)^2 + 4(1-v_3+v_1v_2)}}{2\sqrt{v_1v_2v_3}} \right) \\ &= \ln \left( \frac{(1-x_2^2x_3^2) + \sqrt{(1-x_2^2x_3^2)^2 + 4(1-x_3(1-x_1^2x_2^2))}}{2\sqrt{x_1^2x_2^2x_3^2}} \right) \\ &= \ln \left( \frac{(1-x_2^2x_3^2) + \sqrt{(1-x_2^2x_3^2)^2 + 4(1-x_3(1-x_1^2x_2^2))}}{2x_1x_2x_3} \right) \\ &= \ln \left( \frac{\left(1-e^{-2(\tau_2^{(12)}+\tau_3^{(12)})}\right) + \sqrt{\left(1-e^{-2(\tau_2^{(12)}+\tau_3^{(12)})}\right)^2 + 4\left(1-e^{-\tau_3^{(12)}}\left(1-e^{-2(\tau_1^{(12)}+\tau_2^{(12)})}\right)\right)}}{2e^{-(\tau_1^{(12)}+\tau_2^{(12)}+\tau_3^{(12)})}} \right). \end{aligned} \quad (\text{D.17})$$

Recall that

$$w_3 = y_3 = e^{-\tau_3^{(23)}} \quad (\text{D.18})$$

and

$$w_3 = \frac{1}{2} \left( (1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma} \right). \quad (\text{D.19})$$

Again, using the appropriate substitutions,

$$\begin{aligned} \tau_3^{(23)} &= -\ln \frac{1}{2} \left( (1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma} \right) \\ &= \ln \left( \frac{2}{(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma}} \right) \\ &= \ln \left( \frac{2}{(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma}} \cdot \frac{-(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma}}{-(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma}} \right) \\ &= \ln \left( \frac{-2(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma}}{4\gamma} \right) \\ &= \ln \left( \frac{-(1 - \beta) + \sqrt{(1 - \beta)^2 + 4\gamma}}{2\gamma} \right) \quad (\text{D.20}) \\ &= \ln \left( \frac{-(1 - v_2 v_3) + \sqrt{(1 - v_2 v_3)^2 + 4(1 - v_3 + v_1 v_2)}}{2(1 - v_3 + v_1 v_2)} \right) \\ &= \ln \left( \frac{-(1 - x_2^2 x_3^2) + \sqrt{(1 - x_2^2 x_3^2)^2 + 4(1 - x_3(1 - x_1^2 x_2^2))}}{2(1 - x_3(1 - x_1^2 x_2^2))} \right) \\ &= \ln \left( \frac{-\left(1 - e^{-2(\tau_2^{(12)} + \tau_3^{(12)})}\right) + \sqrt{\left(1 - e^{-2(\tau_2^{(12)} + \tau_3^{(12)})}\right)^2 + 4\left(1 - e^{-\tau_3^{(12)}}\left(1 - e^{-2(\tau_1^{(12)} + \tau_2^{(12)})}\right)\right)}}{2\left(1 - e^{-\tau_3^{(12)}}\left(1 - e^{-2(\tau_1^{(12)} + \tau_2^{(12)})}\right)\right)} \right). \end{aligned}$$

In summary,

$$\begin{cases} \tau_1^{(23)} = \ln \sqrt{\frac{e^{-2(\tau_1^{(12)} + \tau_2^{(12)} + \tau_3^{(12)})}}{1 - e^{-\tau_3^{(12)}} \left(1 - e^{-2(\tau_1^{(12)} + \tau_2^{(12)})}\right)}}, \\ \tau_2^{(23)} = \ln \left( \frac{\left(1 - e^{-2(\tau_2^{(12)} + \tau_3^{(12)})}\right) + \sqrt{\left(1 - e^{-2(\tau_2^{(12)} + \tau_3^{(12)})}\right)^2 + 4 \left(1 - e^{-\tau_3^{(12)}} \left(1 - e^{-2(\tau_1^{(12)} + \tau_2^{(12)})}\right)\right)}}{2e^{-(\tau_1^{(12)} + \tau_2^{(12)} + \tau_3^{(12)})}} \right), \\ \tau_3^{(23)} = \ln \left( \frac{-\left(1 - e^{-2(\tau_2^{(12)} + \tau_3^{(12)})}\right) + \sqrt{\left(1 - e^{-2(\tau_2^{(12)} + \tau_3^{(12)})}\right)^2 + 4 \left(1 - e^{-\tau_3^{(12)}} \left(1 - e^{-2(\tau_1^{(12)} + \tau_2^{(12)})}\right)\right)}}{2 \left(1 - e^{-\tau_3^{(12)}} \left(1 - e^{-2(\tau_1^{(12)} + \tau_2^{(12)})}\right)\right)} \right). \end{cases} \quad (\text{D.21})$$

Recall that

$$\tau_1^{(23)} = \ln \sqrt{\frac{x_1^2 x_2^2 x_3^2}{1 - x_3 (1 - x_1^2 x_2^2)}}. \quad (\text{D.22})$$

Demanding  $0 < x_1, x_2, x_3 \leq 1$  and  $\tau_1^{(23)} \geq 0$ ,

$$\begin{aligned} x_1^2 x_2^2 x_3^2 &\geq 1 - x_3 (1 - x_1^2 x_2^2) \\ \Leftrightarrow x_1^2 x_2^2 x_3^2 &\geq 1 - x_3 + x_1^2 x_2^2 x_3 \\ \Leftrightarrow 0 &\geq 1 - x_3 + x_1^2 x_2^2 x_3 - x_1^2 x_2^2 x_3^2 \\ \Leftrightarrow 0 &\geq 1 - x_3 (1 - x_1^2 x_2^2 + x_1^2 x_2^2 x_3) \\ \Leftrightarrow 0 &\geq 1 - x_3 (1 - x_1^2 x_2^2 (1 - x_3)). \end{aligned} \quad (\text{D.23})$$

This is the same inequality as in (B.26) on page 190, except with the direction of the inequality swapped. We can conclude that  $\tau_1^{(23)} \leq 0$ .

Recall that

$$\begin{aligned}
\tau_2^{(23)} &= \ln \left( \frac{(1-\beta) + \sqrt{(1-\beta)^2 + 4\gamma}}{2\sqrt{\alpha}} \right) \\
&= \ln \left( \frac{(1-\beta) + \sqrt{(1-\beta)^2 + 4\gamma}}{2\sqrt{\alpha}} \cdot \frac{-(1-\beta) + \sqrt{(1-\beta)^2 + 4\gamma}}{-(1-\beta) + \sqrt{(1-\beta)^2 + 4\gamma}} \right) \\
&= \ln \left( \frac{4\gamma}{2\sqrt{\alpha} \left( -(1-\beta) + \sqrt{(1-\beta)^2 + 4\gamma} \right)} \right) \\
&= \ln \left( \frac{2\gamma}{\sqrt{\alpha} \left( -(1-\beta) + \sqrt{(1-\beta)^2 + 4\gamma} \right)} \right).
\end{aligned} \tag{D.24}$$

Demanding  $0 < \alpha, \beta, \gamma \leq 1$  and  $\tau_2^{(23)} \geq 0$ ,

$$\begin{aligned}
&2\gamma \geq \sqrt{\alpha} \left( -(1-\beta) + \sqrt{(1-\beta)^2 + 4\gamma} \right) \\
&\Leftrightarrow \frac{2\gamma}{\sqrt{\alpha}} \geq -(1-\beta) + \sqrt{(1-\beta)^2 + 4\gamma} \\
&\Leftrightarrow (1-\beta) + \frac{2\gamma}{\sqrt{\alpha}} \geq \sqrt{(1-\beta)^2 + 4\gamma} \\
&\Leftrightarrow (1-\beta)^2 + \frac{4\gamma^2}{\alpha} + \frac{4\gamma(1-\beta)}{\sqrt{\alpha}} \geq (1-\beta)^2 + 4\gamma \\
&\Leftrightarrow \frac{4\gamma^2}{\alpha} + \frac{4\gamma(1-\beta)}{\sqrt{\alpha}} \geq 4\gamma \\
&\Leftrightarrow \frac{\gamma}{\alpha} + \frac{(1-\beta)}{\sqrt{\alpha}} \geq 1 \\
&\Leftrightarrow \gamma + \sqrt{\alpha}(1-\beta) \geq \alpha \\
&\Leftrightarrow \sqrt{\alpha}(1-\beta) \geq \alpha - \gamma \\
&\Leftrightarrow \sqrt{x_1^2 x_2^2 x_3^2} (1 - x_2^2 x_3^2) \geq x_1^2 x_2^2 x_3^2 - (1 - x_3 (1 - x_1^2 x_2^2)) \\
&\Leftrightarrow x_1 x_2 x_3 (1 - x_2^2 x_3^2) \geq x_1^2 x_2^2 x_3^2 - 1 + x_3 - x_1^2 x_2^2 x_3 \\
&\Leftrightarrow x_1 x_2 x_3 (1 - x_2^2 x_3^2) \geq -(1 - x_3 + x_1^2 x_2^2 x_3 - x_1^2 x_2^2 x_3^2) \\
&\Leftrightarrow x_1 x_2 x_3 (1 - x_2^2 x_3^2) \geq -(1 - x_3 (1 - x_1^2 x_2^2 + x_1^2 x_2^2 x_3)) \\
&\Leftrightarrow x_1 x_2 x_3 (1 - x_2^2 x_3^2) \geq -(1 - x_3 (1 - x_1^2 x_2^2 (1 - x_3))) .
\end{aligned} \tag{D.25}$$

Since  $0 < x_1, x_2, x_3 \leq 1$ ,

$$\begin{aligned}
& 0 < x_2^2 x_3^2 \leq 1 \\
& \Leftrightarrow -1 \leq -x_2^2 x_3^2 < 0 \\
& \Leftrightarrow 0 \leq 1 - x_2^2 x_3^2 < 1 \\
& \Leftrightarrow 0 \leq x_1 x_2 x_3 (1 - x_2^2 x_3^2) < 1.
\end{aligned} \tag{D.26}$$

From (B.26) on page 190, we can conclude that  $-1 < -(1 - x_3 (1 - x_1^2 x_2^2 (1 - x_3))) \leq 0$ . Since  $0 \leq x_1 x_2 x_3 (1 - x_2^2 x_3^2) < 1$ , we can conclude that  $\tau_2^{(23)} \geq 0$ .

Recall that

$$\tau_3^{(23)} = \ln \left( \frac{-(1 - x_2^2 x_3^2) + \sqrt{(1 - x_2^2 x_3^2)^2 + 4(1 - x_3 (1 - x_1^2 x_2^2))}}{2(1 - x_3 (1 - x_1^2 x_2^2))} \right). \tag{D.27}$$

Demanding  $0 < x_1, x_2, x_3 \leq 1$  and  $\tau_3^{(23)} \geq 0$ ,

$$\begin{aligned}
& -(1 - x_2^2 x_3^2) + \sqrt{(1 - x_2^2 x_3^2)^2 + 4(1 - x_3 (1 - x_1^2 x_2^2))} \geq 2(1 - x_3 (1 - x_1^2 x_2^2)) \\
& \Leftrightarrow \sqrt{(1 - x_2^2 x_3^2)^2 + 4(1 - x_3 (1 - x_1^2 x_2^2))} \geq (1 - x_2^2 x_3^2) + 2(1 - x_3 (1 - x_1^2 x_2^2)) \\
& \Leftrightarrow (1 - x_2^2 x_3^2)^2 + 4(1 - x_3 (1 - x_1^2 x_2^2)) \geq (1 - x_2^2 x_3^2)^2 + 4(1 - x_3 (1 - x_1^2 x_2^2))^2 \\
& \quad + 4(1 - x_2^2 x_3^2)(1 - x_3 (1 - x_1^2 x_2^2)) \\
& \Leftrightarrow 4(1 - x_3 (1 - x_1^2 x_2^2)) \geq 4(1 - x_3 (1 - x_1^2 x_2^2))^2 \\
& \quad + 4(1 - x_2^2 x_3^2)(1 - x_3 (1 - x_1^2 x_2^2)) \\
& \Leftrightarrow 1 \geq (1 - x_3 (1 - x_1^2 x_2^2)) + (1 - x_2^2 x_3^2) \\
& \Leftrightarrow 0 \geq -x_3 + x_1^2 x_2^2 x_3 + 1 - x_2^2 x_3^2 \\
& \Leftrightarrow 0 \geq 1 - x_3 (1 - x_1^2 x_2^2 + x_2^2 x_3) \\
& \Leftrightarrow 0 \geq 1 - x_3 (1 - x_2^2 (x_1^2 - x_3)).
\end{aligned} \tag{D.28}$$

This is the same inequality as in (A.25) on page 181, except with the direction of the inequality swapped. We can conclude that the inequality will not be satisfied when  $x_3 < x_1^2$ . Clearly, when  $x_3 = x_1^2$  the inequality will be satisfied, since  $1 - x_3 > 0$ . We can conclude that  $\tau_3^{(23)}$  can sometimes be non-negative and sometimes be negative.

To get a rough estimate of how often the inequality is satisfied we decided to simulate some samples for  $x_1, x_2, x_3 \in U(0, 1)$ . We found that for 60 out of 500 samples  $\tau_3^{(23)} \geq 0$ . Of the remaining 440 out of 500 samples  $\tau_3^{(23)} < 0$ .



## Appendix E

# Gröbner Basis for Network 1b

Using the *Macaulay2* code from Section 6.1 on page 125, the Gröbner basis of the ideal for Network 1b is

$$\begin{aligned} I = \langle & q_{1010} - q_{1100}, \\ & q_{0101} - q_{0110}, \\ & q_{0011}^2 q_{1100} - q_{0011} q_{0110} q_{1001}^2 + q_{0011} q_{0110} q_{1001} - 2q_{0011} q_{0110} q_{1100} + q_{0011} q_{1001} q_{1111} \\ & - q_{0011} q_{1111} - q_{0110}^2 q_{1001}, \\ & y_4 q_{0110} q_{1001} - y_4 q_{1111} + q_{0011} q_{1100} - q_{0110} q_{1100} + q_{0110}^2 q_{1100} + q_{0110} q_{1001} q_{1111} \\ & + q_{0110} q_{1111} - q_{1111}^2, \\ & y_4 q_{0011} - y_4 q_{0110} + q_{0011} q_{1001} - q_{0011} + q_{0110} - q_{1111}, \\ & y_4^2 + y_4 q_{1001} - y_4 - q_{1100}, \\ & y_3^2 q_{1100}^2 + y_4 q_{0110} - y_4 q_{1111} + q_{0110} q_{1001} - q_{0110} q_{1100} - q_{0110} - q_{1001} q_{1111} + q_{1111}, \\ & y_3^2 q_{0011} q_{1001} q_{1100} - y_3^2 q_{0011} q_{1100} + y_3^2 q_{0110} q_{1100} - y_3^2 q_{1100} q_{1111} - q_{0011} q_{0110} q_{1001} \\ & + q_{0011} q_{1111}, \\ & y_3^2 q_{0011} q_{1001}^2 - 2y_3^2 q_{0011} q_{1001} + y_3^2 q_{0011} q_{1100} + y_3^2 q_{0011} + y_3^2 q_{0110} q_{1001} - y_3^2 q_{0110} q_{1100} \\ & - y_3^2 q_{0110} - y_3^2 q_{1001} q_{1111} + y_3^2 q_{1111} - q_{0011}^2 + q_{0011} q_{0110}, \\ & y_3^2 y_4 q_{1100} - y_4 q_{0110} + q_{0110} - q_{1111}, \\ & y_3^2 y_4 q_{1001} - y_3^2 y_4 - y_3^2 q_{1100} + q_{0011}, \\ & y_3^2 y_4 q_{0110} - y_3^2 y_4 q_{1111} + y_3^2 q_{0011} q_{1100} - q_{0011} q_{0110}, \\ & y_2^2 q_{0110}^2 q_{1001} - y_2^2 q_{0110}^2 q_{1100} - y_2^2 q_{0110} q_{1001} q_{1111} - y_2^2 q_{0110} q_{1111} + y_2^2 q_{1111}^2 \\ & - q_{0011} q_{0110} q_{1100} + q_{0110}^2 q_{1001}^2 - q_{0110}^2 q_{1001} + 2q_{0110}^2 q_{1100} - q_{0110} q_{1001} q_{1111} + q_{0110} q_{1111}, \\ & y_2^2 q_{0011} - q_{0110}, \\ & y_2^2 y_4 q_{1111} - y_2^2 q_{0110} q_{1001} + y_2^2 q_{0110} q_{1100} + y_2^2 q_{1001} q_{1111} - y_4 q_{1111} + q_{0011} q_{1100} \end{aligned}$$

$$\begin{aligned}
& -q_{0110}q_{1001}^2 + q_{0110}q_{1001} - 2q_{0110}q_{1100}, \\
& y_2^2y_4q_{0110} - y_2^2q_{0110} + y_2^2q_{1111} - y_4q_{0110} - q_{0110}q_{1001} + q_{0110}, \\
& y_2^2y_3^2q_{0110}q_{1100} - y_2^2y_3^2q_{1100}q_{1111} + y_3^2q_{0110}q_{1001}q_{1100} - y_3^2q_{0110}q_{1100} - q_{0110}^2q_{1001} \\
& + q_{0110}q_{1111}, \\
& y_2^2y_3^2q_{0110}q_{1001} - y_2^2y_3^2q_{0110} - y_2^2y_3^2q_{1001}q_{1111} - y_2^2y_3^2q_{1100}q_{1111} + y_2^2y_3^2q_{1111} + y_3^2q_{0110}q_{1001}^2 \\
& + y_3^2q_{0110}q_{1001}q_{1100} - 2y_3^2q_{0110}q_{1001} + y_3^2q_{0110} - q_{0011}q_{0110} - q_{0110}^2q_{1001} + q_{0110}^2 \\
& + q_{0110}q_{1111}, \\
& y_2^2y_3^2q_{0110}^2 - 2y_2^2y_3^2q_{0110}q_{1111} + y_2^2y_3^2q_{1111}^2 - y_3^2q_{0011}q_{0110}q_{1100} + y_3^2q_{0110}^2q_{1001} \\
& + y_3^2q_{0110}^2q_{1100} - y_3^2q_{0110}^2 - y_3^2q_{0110}q_{1001}q_{1111} + y_3^2q_{0110}q_{1111} + q_{0011}q_{0110}^2 - q_{0110}^3, \\
& y_1^2q_{0110} - q_{1100}, \\
& y_1^2y_4q_{1111} - y_1^2q_{0011}q_{1100} - y_4q_{1001}q_{1100} + q_{1100}^2, \\
& y_1^2y_2^2q_{1111} + y_2^2y_4q_{1100} - y_2^2q_{1100} - y_4q_{1100} - q_{1001}q_{1100} + q_{1100}, \\
& y_1^2y_2^2y_3^2q_{1100} - y_4q_{1001} + y_4 - q_{1001}^2 + 2q_{1001} - q_{1100} - 1, \\
& y_1^2y_2^2y_3^2y_4 - y_4 - q_{1001} + 1 \rangle.
\end{aligned}$$

## Appendix F

# Gröbner Basis for Network 1c

Using the *Macaulay2* code from Section 6.1 on page 125, the Gröbner basis of the ideal for Network 1c is

$$\begin{aligned} I = \langle & q_{1010} - q_{1100}, \\ & q_{1001} - q_{1100}, \\ & q_{0011}^2 q_{1100}^2 - q_{0011} q_{0101}^2 q_{1100}^2 + q_{0011} q_{0101} q_{1100}^2 + q_{0011} q_{0101} q_{1100} q_{1111} - 2q_{0011} q_{0110} q_{1100}^2 \\ & - q_{0011} q_{1100} q_{1111} - q_{0101} q_{0110} q_{1100}^2 + q_{0101} q_{0110} q_{1100} q_{1111} + q_{0110}^2 q_{1100}^2 + q_{0110} q_{1100} q_{1111} \\ & - q_{0110} q_{1111}^2, \\ & y_4 q_{0101} q_{1100} - y_4 q_{1111} + q_{0011} q_{1100} - q_{0110} q_{1100}, \\ & y_4 q_{0011} q_{1111} - y_4 q_{0110} q_{1111} - q_{0011}^2 q_{1100} + q_{0011} q_{0101}^2 q_{1100} - q_{0011} q_{0101} q_{1100} \\ & + 2q_{0011} q_{0110} q_{1100} + q_{0101} q_{0110} q_{1100} - q_{0101} q_{0110} q_{1111} - q_{0110}^2 q_{1100}, \\ & y_4 q_{0011} q_{1100} - y_4 q_{0110} q_{1100} + q_{0011} q_{0101} q_{1100} - q_{0011} q_{1100} + q_{0110} q_{1100} - q_{0110} q_{1111}, \\ & y_4^2 + y_4 q_{0101} - y_4 - q_{0110}, \\ & y_3^2 q_{0110} q_{1100} + y_4 q_{1100} - y_4 q_{1111} + q_{0101} q_{1100} - q_{0101} q_{1111} - q_{0110} q_{1100} - q_{1100} + q_{1111}, \\ & y_3^2 q_{0110}^2 - y_4 q_{0011} q_{0101} + y_4 q_{0011} - q_{0011} q_{0101}^2 + 2q_{0011} q_{0101} - q_{0011} q_{0110} - q_{0011}, \\ & y_3^2 q_{0011} q_{1100}^2 + y_4 q_{1100}^2 - y_4 q_{1100} q_{1111} - q_{0011} q_{1100}^2 - q_{1100}^2 + 2q_{1100} q_{1111} - q_{1111}^2, \\ & y_3^2 q_{0011} q_{0101} q_{1100} - y_3^2 q_{0011} q_{1100} - y_3^2 q_{0110} q_{1111} - y_4 q_{1100} + y_4 q_{1111} - q_{0011} q_{0101} q_{1100} \\ & + q_{0011} q_{1111} - q_{0101} q_{1100} + q_{0101} q_{1111} + q_{0110} q_{1100} + q_{1100} - q_{1111}, \\ & y_3^2 y_4 q_{1111} - y_3^2 q_{0011} q_{1100} - y_4 q_{1100} + q_{0011} q_{1100} + q_{1100} - q_{1111}, \\ & y_3^2 y_4 q_{1100} - y_4 q_{1100} + q_{1100} - q_{1111}, \\ & y_3^2 y_4 q_{0110} - y_4 q_{0011} - q_{0011} q_{0101} + q_{0011}, \\ & y_3^2 y_4 q_{0101} - y_3^2 y_4 - y_3^2 q_{0110} + q_{0011}, \\ & y_2^2 q_{0101} q_{1100}^2 - y_2^2 q_{0101} q_{1100} q_{1111} - y_2^2 q_{0110} q_{1100}^2 - y_2^2 q_{1100} q_{1111} + y_2^2 q_{1111}^2 - q_{0011} q_{1100}^2 \end{aligned}$$

$$\begin{aligned}
& +q_{0101}^2q_{1100}^2 - q_{0101}q_{1100}^2 - q_{0101}q_{1100}q_{1111} + 2q_{0110}q_{1100}^2 + q_{1100}q_{1111}, \\
& y_2^2q_{0011} - q_{0110}, \\
& y_2^2y_4q_{1111} - y_2^2q_{0101}q_{1100} + y_2^2q_{0101}q_{1111} + y_2^2q_{0110}q_{1100} - y_4q_{1111} + q_{0011}q_{1100} \\
& - q_{0101}^2q_{1100} + q_{0101}q_{1100} - 2q_{0110}q_{1100}, \\
& y_2^2y_4q_{1100} - y_2^2q_{1100} + y_2^2q_{1111} - y_4q_{1100} - q_{0101}q_{1100} + q_{1100}, \\
& y_2^2y_3^2q_{1100} - y_2^2y_3^2q_{1111} + y_3^2q_{0101}q_{1100} - y_3^2q_{1100} - q_{0101}q_{1100} + q_{1111}, \\
& y_2^2y_3^2q_{0110} - y_4q_{0101} + y_4 - q_{0101}^2 + 2q_{0101} - q_{0110} - 1, \\
& y_2^2y_3^2y_4 - y_4 - q_{0101} + 1, \\
& y_1^2q_{0110} - q_{1100}, \\
& y_1^2q_{0011}q_{1100} - y_1^2q_{0101}^2q_{1100} + y_1^2q_{0101}q_{1100} + y_1^2q_{0101}q_{1111} - y_1^2q_{1111} + y_2^2y_3^2q_{1111}^2 \\
& - y_3^2q_{0101}q_{1100}^2 - y_3^2q_{0101}q_{1100}q_{1111} + y_3^2q_{1100}q_{1111} + q_{0101}q_{1100}^2 + q_{0101}q_{1100}q_{1111} - q_{1100}^2 \\
& - q_{1100}q_{1111} - q_{1111}^2, \\
& y_1^2y_4q_{1100} - y_1^2y_4q_{1111} + y_1^2q_{0101}q_{1100} - y_1^2q_{0101}q_{1111} - y_1^2q_{1100} + y_1^2q_{1111} \\
& + y_3^2q_{1100}^2 - q_{1100}^2, \\
& y_1^2y_4q_{0101} - y_1^2y_4 + y_1^2q_{0101}^2 - 2y_1^2q_{0101} + y_1^2 - y_2^2y_3^2q_{1111} + y_3^2q_{0101}q_{1100} - y_3^2q_{1100} \\
& - q_{0101}q_{1100} + q_{1100} + q_{1111}, \\
& y_1^2y_4q_{0011} + y_1^2q_{0011}q_{0101} - y_1^2q_{0011} - y_4q_{1100} + q_{1100} - q_{1111}, \\
& y_1^2y_3^2q_{0101}q_{1100}^2 - y_1^2y_3^2q_{0101}q_{1100}q_{1111} - y_1^2y_3^2q_{1100}^2 + y_1^2y_3^2q_{1100}q_{1111} - y_1^2q_{0101}q_{1100}^2 \\
& + y_1^2q_{0101}q_{1100}q_{1111} + y_1^2q_{1100}q_{1111} - y_1^2q_{1111}^2 + y_3^4q_{1100}^3 - 2y_3^2q_{1100}^3 + q_{1100}^3, \\
& y_1^2y_3^2q_{0101}^2 - 2y_1^2y_3^2q_{0101} + y_1^2y_3^2 - y_1^2q_{0011} - y_2^2y_3^4q_{1111} + y_3^4q_{0101}q_{1100} - y_3^4q_{1100} \\
& - y_3^2q_{0101}q_{1100} + 2y_3^2q_{1100} + y_3^2q_{1111}, \\
& y_1^2y_2^2q_{0101}^2q_{1100} - y_1^2y_2^2q_{0101}q_{1100} - y_1^2y_2^2q_{0101}q_{1111} + y_1^2y_2^2q_{1111} - y_2^4y_3^2q_{1111}^2 \\
& + 2y_2^2y_3^2q_{0101}q_{1111}^2 - y_2^2y_3^2q_{1111}^2 - 2y_2^2q_{0101}q_{1100}q_{1111} - y_2^2q_{0110}q_{1100}^2 + y_2^2q_{1100}^2 + 2y_2^2q_{1111}^2 \\
& - y_3^2q_{0101}^2q_{1100}^2 - 2y_3^2q_{0101}^2q_{1100}q_{1111} + y_3^2q_{0101}q_{1100}^2 + 3y_3^2q_{0101}q_{1100}q_{1111} - y_3^2q_{1100}q_{1111} \\
& - q_{0011}q_{1100}^2 + 2q_{0101}^2q_{1100}^2 + 2q_{0101}^2q_{1100}q_{1111} - q_{0101}q_{1100}^2 - 3q_{0101}q_{1100}q_{1111} \\
& - 2q_{0101}q_{1111}^2 + 2q_{0110}q_{1100}^2 - q_{1100}^2 + q_{1100}q_{1111} + q_{1111}^2 \rangle.
\end{aligned}$$

## Appendix G

# Gröbner Basis for Network 2b

Using the *Macaulay2* code from Section 6.1 on page 125, the Gröbner basis of the ideal for Network 2b is

$$\begin{aligned} I = & \langle q_{0110} - q_{1001}, \\ & q_{0101} - q_{1001}, \\ & q_{0011}^2 q_{1100}^2 - 2q_{0011} q_{1001}^2 q_{1100} - q_{0011} q_{1001} q_{1010}^2 q_{1100} + q_{0011} q_{1001} q_{1010} q_{1100} \\ & + q_{0011} q_{1010} q_{1100} q_{1111} - q_{0011} q_{1100} q_{1111} + q_{1001}^4 - q_{1001}^3 q_{1010} + q_{1001}^2 q_{1010} q_{1111} \\ & + q_{1001}^2 q_{1111} - q_{1001} q_{1111}^2, \\ & y_4 q_{1001} q_{1010} - y_4 q_{1111} + q_{0011} q_{1100} - q_{1001}^2, \\ & y_4 q_{0011} q_{1100} - y_4 q_{1001}^2 + q_{0011} q_{1010} q_{1100} - q_{0011} q_{1100} + q_{1001}^2 - q_{1001} q_{1111}, \\ & y_4^2 + y_4 q_{1010} - y_4 - q_{1001}, \\ & y_3^2 q_{1001} q_{1100} + y_4 q_{1001} - y_4 q_{1111} - q_{1001}^2 + q_{1001} q_{1010} - q_{1001} - q_{1010} q_{1111} + q_{1111}, \\ & y_3^2 q_{1001}^2 - y_4 q_{0011} q_{1010} + y_4 q_{0011} - q_{0011} q_{1001} - q_{0011} q_{1010}^2 + 2q_{0011} q_{1010} - q_{0011}, \\ & y_3^2 q_{0011} q_{1100}^2 + y_4 q_{1001}^2 - y_4 q_{1001} q_{1111} - q_{0011} q_{1001} q_{1100} - q_{1001}^2 + 2q_{1001} q_{1111} - q_{1111}^2, \\ & y_3^2 q_{0011} q_{1010} q_{1100} - y_3^2 q_{0011} q_{1100} - y_3^2 q_{1001} q_{1111} + y_4 q_{0011} q_{1010} - y_4 q_{0011} \\ & - q_{0011} q_{1001} q_{1010} + q_{0011} q_{1001} + q_{0011} q_{1010}^2 - 2q_{0011} q_{1010} + q_{0011} q_{1111} + q_{0011}, \\ & y_3^2 y_4 q_{1111} - y_3^2 q_{0011} q_{1100} - y_4 q_{0011} + q_{0011} q_{1001} - q_{0011} q_{1010} + q_{0011}, \\ & y_3^2 y_4 q_{1100} - y_4 q_{1001} + q_{1001} - q_{1111}, \\ & y_3^2 y_4 q_{1010} - y_3^2 y_4 - y_3^2 q_{1001} + q_{0011}, \\ & y_3^2 y_4 q_{1001} - y_4 q_{0011} - q_{0011} q_{1010} + q_{0011}, \\ & y_2^2 q_{1001}^3 - y_2^2 q_{1001}^2 q_{1010} + y_2^2 q_{1001} q_{1010} q_{1111} + y_2^2 q_{1001} q_{1111} - y_2^2 q_{1111}^2 + y_3^2 q_{1100}^3 \\ & - y_4 q_{1010} q_{1100}^2 + y_4 q_{1100}^2 - 2q_{1001} q_{1100}^2 - q_{1010}^2 q_{1100}^2 + 2q_{1010} q_{1100}^2 - q_{1100}^2, \\ & y_2^2 q_{0011} - q_{1100}, \end{aligned}$$

$$\begin{aligned}
& y_2^2 y_4 q_{1111}^2 + y_2^2 q_{1001}^2 q_{1111} - y_2^2 q_{1001} q_{1010} q_{1111} + y_2^2 q_{1010} q_{1111}^2 - y_3^2 q_{1010} q_{1100}^3 \\
& - y_4 q_{1010} q_{1100}^2 + q_{1001} q_{1010} q_{1100}^2 - q_{1010}^2 q_{1100}^2 + q_{1010} q_{1100}^2 - q_{1100}^2 q_{1111}, \\
& y_2^2 y_4 q_{1001} q_{1111} - y_2^2 q_{1001} q_{1111} + y_2^2 q_{1111}^2 - y_3^2 q_{1100}^3 - y_4 q_{1100}^2 + q_{1001} q_{1100}^2 - q_{1010} q_{1100}^2 \\
& + q_{1100}^2, \\
& y_2^2 y_4 q_{1001}^2 - y_2^2 q_{1001}^2 + y_2^2 q_{1001} q_{1111} - y_4 q_{1100}^2 - q_{1010} q_{1100}^2 + q_{1100}^2, \\
& y_2^2 y_3^2 q_{1111}^2 - y_3^4 q_{1100}^3 - 2y_3^2 q_{1010} q_{1100}^2 + 2y_3^2 q_{1100}^2 - 2y_4 q_{1001} q_{1100} - y_4 q_{1010} q_{1100} \\
& + 2y_4 q_{1100} q_{1111} + y_4 q_{1100} + q_{1001}^2 q_{1100} + q_{1001} q_{1100} - q_{1010}^2 q_{1100} + 2q_{1010} q_{1100} q_{1111} \\
& + 2q_{1010} q_{1100} - 4q_{1100} q_{1111} - q_{1100}, \\
& y_2^2 y_3^2 q_{1001} q_{1111} - y_3^2 q_{1010} q_{1100}^2 + y_3^2 q_{1100}^2 - y_4 q_{1010} q_{1100} + y_4 q_{1100} + q_{1001} q_{1010} q_{1100} \\
& - q_{1001} q_{1100} - q_{1010}^2 q_{1100} + 2q_{1010} q_{1100} - q_{1100} q_{1111} - q_{1100}, \\
& y_1^2 q_{1100} - q_{1001}, \\
& y_1^2 q_{1001}^3 - y_1^2 q_{1001}^2 q_{1010} + y_1^2 q_{1001} q_{1010} q_{1111} + y_1^2 q_{1001} q_{1111} - y_1^2 q_{1111}^2 + q_{0011}^2 q_{1100} \\
& - 2q_{0011} q_{1001}^2 - q_{0011} q_{1001} q_{1010}^2 + q_{0011} q_{1001} q_{1010} + q_{0011} q_{1010} q_{1111} - q_{0011} q_{1111}, \\
& y_1^2 y_4 q_{1111} + y_1^2 q_{1001}^2 - y_1^2 q_{1001} q_{1010} + y_1^2 q_{1010} q_{1111} - y_4 q_{0011} q_{1010} - q_{0011} q_{1001} \\
& - q_{0011} q_{1010}^2 + q_{0011} q_{1010}, \\
& y_1^2 y_4 q_{1001} - y_1^2 q_{1001} + y_1^2 q_{1111} - y_4 q_{0011} - q_{0011} q_{1010} + q_{0011}, \\
& y_1^2 y_3^2 q_{1111}^2 - y_1^2 y_4 q_{0011} q_{1010} + y_1^2 y_4 q_{0011} - y_1^2 q_{0011} q_{1001}^2 + 2y_1^2 q_{0011} q_{1001} q_{1010} \\
& - y_1^2 q_{0011} q_{1001} - y_1^2 q_{0011} q_{1010}^2 + 2y_1^2 q_{0011} q_{1010} - 2y_1^2 q_{0011} q_{1111} - y_1^2 q_{0011} - y_3^2 q_{0011}^2 q_{1100} \\
& - y_3^2 q_{0011} q_{1001} q_{1010} + y_3^2 q_{0011} q_{1001} - y_3^2 q_{0011} q_{1010} q_{1111} + y_3^2 q_{0011} q_{1111} + y_4 q_{0011}^2 q_{1010} \\
& - y_4 q_{0011}^2 + 2q_{0011}^2 q_{1001} + q_{0011}^2 q_{1010} - 2q_{0011}^2 q_{1010} + q_{0011}^2, \\
& y_1^2 y_3^2 q_{1001} - y_1^2 y_3^2 q_{1111} - y_1^2 y_4 q_{0011} - y_1^2 q_{0011} q_{1010} + y_1^2 q_{0011} + y_3^2 y_4 q_{0011} + y_3^2 q_{0011} q_{1010} \\
& - y_3^2 q_{0011}, \\
& y_1^2 y_2^2 y_3^2 q_{1111} - y_3^2 q_{1010} q_{1100} + y_3^2 q_{1100} - y_4 q_{1010} + y_4 + q_{1001} q_{1010} - q_{1001} - q_{1010}^2 \\
& + 2q_{1010} - q_{1111} - 1, \\
& y_1^2 y_2^2 y_3^2 y_4 - y_4 - q_{1010} + 1 \rangle.
\end{aligned}$$

## Appendix H

# Gröbner Basis for Network 3

Using the *Macaulay2* code from Section 6.1 on page 125, the Gröbner basis of the ideal for Network 3 is

$$\begin{aligned} I = \langle & q_{0101}q_{1010} - q_{0110}q_{1001}, \\ & q_{0011}q_{1100} - q_{1111}, \\ & y_{1234}^2 q_{1111} - q_{0110}q_{1001}, \\ & y_4^2 q_{1010} - q_{0011}q_{1001}, \\ & y_4^2 q_{0110} - q_{0011}q_{0101}, \\ & y_4^2 y_{1234}^2 q_{1100} - q_{0101}q_{1001}, \\ & y_3 q_{1001} - y_4 q_{1010}, \\ & y_3 q_{0101} - y_4 q_{0110}, \\ & y_3 y_4 - q_{0011}, \\ & y_3^2 y_{1234}^2 q_{1100} - q_{0110}q_{1010}, \\ & y_2 q_{1010} - y_3 y_{1234} q_{1100}, \\ & y_2 q_{1001} - y_4 y_{1234} q_{1100}, \\ & y_2 y_{1234} q_{1111} - y_4 q_{0110} q_{1100}, \\ & y_2 y_{1234} q_{0011} - y_4 q_{0110}, \\ & y_2 y_4 y_{1234} - q_{0101}, \\ & y_2 y_3 y_{1234} - q_{0110}, \\ & y_1 q_{0110} - y_3 y_{1234} q_{1100}, \\ & y_1 q_{0101} - y_4 y_{1234} q_{1100}, \\ & y_1 y_{1234} q_{1111} - y_4 q_{1010} q_{1100}, \\ & y_1 y_{1234} q_{0011} - y_4 q_{1010}, \end{aligned}$$

$$y_1 y_4 y_{1234} - q_{1001},$$

$$y_1 y_3 y_{1234} - q_{1010},$$

$$y_1 y_2 - q_{1100} \rangle.$$



## Appendix I

# Regions in the Probability Space

**Region**  $R_1 \cap R_2 \cap R_3 \cap R_4 \cap R_5$

From (6.95),

$$\begin{aligned} R_1 \cap R_2 \cap R_3 \cap R_4 \cap R_5 &= \Omega_1 \cap (\Omega_1 \cap \Omega_2) \cap (\Omega_{1a} \cap \Omega_3) \cap (\Omega_{1a} \cap \Omega_2) \cap \Omega_2 \\ &= \Omega_1 \cap \Omega_{1a} \cap \Omega_2 \cap \Omega_3. \end{aligned} \quad (\text{I.1})$$

From (6.96),

$$\Omega_1 = \Omega_1 \cap \Omega_{1a} \cap \Omega_{1b} \cap \Omega_{1c} \cap \Omega_3. \quad (\text{I.2})$$

From (6.97),

$$\Omega_2 = \Omega_2 \cap \Omega_{2b} \cap \Omega_3. \quad (\text{I.3})$$

We can conclude that

$$\begin{aligned} R_1 \cap R_2 \cap R_3 \cap R_4 \cap R_5 &= (\Omega_1 \cap \Omega_{1a} \cap \Omega_{1b} \cap \Omega_{1c} \cap \Omega_3) \cap (\Omega_2 \cap \Omega_{2b} \cap \Omega_3) \cap \Omega_3 \\ &= \Omega_1 \cap \Omega_{1a} \cap \Omega_{1b} \cap \Omega_{1c} \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3. \end{aligned} \quad (\text{I.4})$$

**Region**  $R_1 \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C$

From (6.95),

$$\begin{aligned} R_1 \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C &= \Omega_1 \cap (\Omega_1 \cap \Omega_{2b})^C \cap (\Omega_{1a} \cap \Omega_3) \cap (\Omega_{1a} \cap \Omega_{2b})^C \cap \Omega_2^C \\ &= \Omega_1 \cap \Omega_{1a} \cap \Omega_2^C \cap \Omega_3 \cap (\Omega_1 \cap \Omega_{2b})^C \cap (\Omega_{1a} \cap \Omega_{2b})^C. \end{aligned} \quad (\text{I.5})$$

For any two sets,  $A$  and  $B$ ,

$$(A \cap B)^C = A^C \cup B^C. \quad (\text{I.6})$$

From (I.6),

$$R_1 \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C = \Omega_1 \cap \Omega_{1a} \cap \Omega_2^C \cap \Omega_3 \cap (\Omega_1^C \cup \Omega_{2b}^C) \cap (\Omega_{1a}^C \cup \Omega_{2b}^C). \quad (\text{I.7})$$

For any three sets,  $A$ ,  $B$  and  $C$ ,

$$(A \cup B) \cap (A \cup C) = A \cup (B \cap C). \quad (\text{I.8})$$

From (I.8),

$$R_1 \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C = \Omega_1 \cap \Omega_{1a} \cap \Omega_2^C \cap \Omega_3 \cap (\Omega_{2b}^C \cup (\Omega_1^C \cap \Omega_{1a}^C)). \quad (\text{I.9})$$

For any three sets,  $A$ ,  $B$  and  $C$ ,

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C). \quad (\text{I.10})$$

From (I.10),

$$R_1 \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C = (\Omega_1 \cap \Omega_{1a} \cap \Omega_2^C \cap \Omega_3 \cap \Omega_{2b}^C) \cup (\Omega_1 \cap \Omega_{1a} \cap \Omega_2^C \cap \Omega_3 \cap \Omega_1^C \cap \Omega_{1a}^C). \quad (\text{I.11})$$

Since  $\Omega_{1a} \cap \Omega_{1a}^C = \emptyset$ ,

$$\begin{aligned} R_1 \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C &= (\Omega_1 \cap \Omega_{1a} \cap \Omega_2^C \cap \Omega_3 \cap \Omega_{2b}^C) \cup \emptyset \\ &= \Omega_1 \cap \Omega_{1a} \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3. \end{aligned} \quad (\text{I.12})$$

From (6.96),

$$\begin{aligned} R_1 \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C &= (\Omega_1 \cap \Omega_{1a} \cap \Omega_{1b} \cap \Omega_{1c} \cap \Omega_3) \cap \Omega_{1a} \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3 \\ &= \Omega_1 \cap \Omega_{1a} \cap \Omega_{1b} \cap \Omega_{1c} \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3. \end{aligned} \quad (\text{I.13})$$

**Region**  $R_1^C \cap R_2^C \cap R_3 \cap R_4 \cap R_5$

From (6.95),

$$\begin{aligned} R_1^C \cap R_2^C \cap R_3 \cap R_4 \cap R_5 &= \Omega_1^C \cap (\Omega_{1b} \cap \Omega_2)^C \cap (\Omega_{1a} \cap \Omega_3) \cap (\Omega_{1a} \cap \Omega_{2b}) \cap \Omega_2 \\ &= \Omega_1^C \cap \Omega_{1a} \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3 \cap (\Omega_{1b} \cap \Omega_2)^C. \end{aligned} \quad (\text{I.14})$$

From (I.6),

$$R_1^C \cap R_2^C \cap R_3 \cap R_4 \cap R_5 = \Omega_1^C \cap \Omega_{1a} \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3 \cap (\Omega_{1b}^C \cup \Omega_2^C). \quad (\text{I.15})$$

From (I.10),

$$\begin{aligned} R_1^C \cap R_2^C \cap R_3 \cap R_4 \cap R_5 \\ = (\Omega_1^C \cap \Omega_{1a} \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3 \cap \Omega_{1b}^C) \cup (\Omega_1^C \cap \Omega_{1a} \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3 \cap \Omega_2^C). \end{aligned} \quad (\text{I.16})$$

Since  $\Omega_2 \cap \Omega_2^C = \emptyset$ ,

$$\begin{aligned} R_1^C \cap R_2^C \cap R_3 \cap R_4 \cap R_5 &= (\Omega_1^C \cap \Omega_{1a} \cap \Omega_{1b}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3) \cup \emptyset \\ &= \Omega_1^C \cap \Omega_{1a} \cap \Omega_{1b}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3. \end{aligned} \quad (\text{I.17})$$

Since  $(R_1^C \cap R_2^C \cap R_3 \cap R_4 \cap R_5) \cap R_2^C = R_1^C \cap R_2^C \cap R_3 \cap R_4 \cap R_5$ , from (6.95),

$$R_1^C \cap R_2^C \cap R_3 \cap R_4 \cap R_5 = (\Omega_1^C \cap \Omega_{1a} \cap \Omega_{1b}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3) \cap (\Omega_{1c} \cap \Omega_2)^C. \quad (\text{I.18})$$

From (I.6),

$$R_1^C \cap R_2^C \cap R_3 \cap R_4 \cap R_5 = (\Omega_1^C \cap \Omega_{1a} \cap \Omega_{1b}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3) \cap (\Omega_{1c}^C \cup \Omega_2^C). \quad (\text{I.19})$$

From (I.10),

$$\begin{aligned} R_1^C \cap R_2^C \cap R_3 \cap R_4 \cap R_5 \\ = (\Omega_1^C \cap \Omega_{1a} \cap \Omega_{1b}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3 \cap \Omega_{1c}^C) \cup (\Omega_1^C \cap \Omega_{1a} \cap \Omega_{1b}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3 \cap \Omega_2^C). \end{aligned} \quad (\text{I.20})$$

Since  $\Omega_2 \cap \Omega_2^C = \emptyset$ ,

$$\begin{aligned} R_1^C \cap R_2^C \cap R_3 \cap R_4 \cap R_5 &= (\Omega_1^C \cap \Omega_{1a} \cap \Omega_{1b}^C \cap \Omega_{1c}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3) \cup \emptyset \\ &= \Omega_1^C \cap \Omega_{1a} \cap \Omega_{1b}^C \cap \Omega_{1c}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3. \end{aligned} \quad (\text{I.21})$$

**Region**  $R_1^C \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C$

From (6.95),

$$\begin{aligned} R_1^C \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C &= \Omega_1^C \cap (\Omega_{1b} \cap \Omega_{2b})^C \cap (\Omega_{1a} \cap \Omega_3) \cap (\Omega_{1a} \cap \Omega_{2b})^C \cap \Omega_2^C \\ &= \Omega_1^C \cap \Omega_{1a} \cap \Omega_2^C \cap \Omega_3 \cap (\Omega_{1b} \cap \Omega_{2b})^C \cap (\Omega_{1a} \cap \Omega_{2b})^C. \end{aligned} \quad (\text{I.22})$$

From (I.6),

$$R_1^C \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C = \Omega_1^C \cap \Omega_{1a} \cap \Omega_2^C \cap \Omega_3 \cap (\Omega_{1b}^C \cup \Omega_{2b}^C) \cap (\Omega_{1a}^C \cup \Omega_{2b}^C). \quad (\text{I.23})$$

From (I.8),

$$R_1^C \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C = \Omega_1^C \cap \Omega_{1a} \cap \Omega_2^C \cap \Omega_3 \cap (\Omega_{2b}^C \cup (\Omega_{1b}^C \cap \Omega_{1a}^C)). \quad (\text{I.24})$$

From (I.10),

$$\begin{aligned} R_1^C \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C \\ = (\Omega_1^C \cap \Omega_{1a} \cap \Omega_2^C \cap \Omega_3 \cap \Omega_{2b}^C) \cup (\Omega_1^C \cap \Omega_{1a} \cap \Omega_2^C \cap \Omega_3 \cap \Omega_{1b}^C \cup \Omega_{1a}^C). \end{aligned} \quad (\text{I.25})$$

Since  $\Omega_{1a} \cap \Omega_{1a}^C = \emptyset$ ,

$$\begin{aligned} R_1^C \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C &= (\Omega_1^C \cap \Omega_{1a} \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3) \cup \emptyset \\ &= \Omega_1^C \cap \Omega_{1a} \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3. \end{aligned} \quad (\text{I.26})$$

Since  $(R_1^C \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C) \cap R_1^C = R_1^C \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C$ , from (6.95),

$$R_1^C \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C = (\Omega_1^C \cap \Omega_{1a} \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3) \cap (\Omega_{1b} \cap \Omega_3)^C. \quad (\text{I.27})$$

From (I.6),

$$R_1^C \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C = (\Omega_1^C \cap \Omega_{1a} \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3) \cap (\Omega_{1b}^C \cup \Omega_3^C). \quad (\text{I.28})$$

From (I.10),

$$\begin{aligned} R_1^C \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C \\ = (\Omega_1^C \cap \Omega_{1a} \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3 \cap \Omega_{1b}^C) \cup (\Omega_1^C \cap \Omega_{1a} \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3 \cap \Omega_3^C). \end{aligned} \quad (\text{I.29})$$

Since  $\Omega_3 \cap \Omega_3^C = \emptyset$ ,

$$\begin{aligned} R_1^C \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C &= (\Omega_1^C \cap \Omega_{1a} \cap \Omega_{1b}^C \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3) \cup \emptyset \\ &= \Omega_1^C \cap \Omega_{1a} \cap \Omega_{1b}^C \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3. \end{aligned} \quad (\text{I.30})$$

Since  $(R_1^C \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C) \cap R_1^C = R_1^C \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C$ , from (6.95),

$$R_1^C \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C = (\Omega_1^C \cap \Omega_{1a} \cap \Omega_{1b}^C \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3) \cap (\Omega_{1c} \cap \Omega_3)^C. \quad (\text{I.31})$$

From (I.6),

$$R_1^C \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C = (\Omega_1^C \cap \Omega_{1a} \cap \Omega_{1b}^C \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3) \cap (\Omega_{1c}^C \cup \Omega_3^C). \quad (\text{I.32})$$

From (I.10),

$$\begin{aligned} R_1^C \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C \\ = (\Omega_1^C \cap \Omega_{1a} \cap \Omega_{1b}^C \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3 \cap \Omega_{1c}^C) \cup (\Omega_1^C \cap \Omega_{1a} \cap \Omega_{1b}^C \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3 \cap \Omega_3^C). \end{aligned} \quad (\text{I.33})$$

Since  $\Omega_3 \cap \Omega_3^C = \emptyset$ ,

$$\begin{aligned} R_1^C \cap R_2^C \cap R_3 \cap R_4^C \cap R_5^C &= (\Omega_1^C \cap \Omega_{1a} \cap \Omega_{1b}^C \cap \Omega_{1c}^C \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3) \cup \emptyset \\ &= \Omega_1^C \cap \Omega_{1a} \cap \Omega_{1b}^C \cap \Omega_{1c}^C \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3. \end{aligned} \quad (\text{I.34})$$

**Region**  $R_1^C \cap R_2^C \cap R_3^C \cap R_4^C \cap R_5$

From (6.95),

$$\begin{aligned} R_1^C \cap R_2^C \cap R_3^C \cap R_4^C \cap R_5 &= \Omega_1^C \cap (\Omega_{1b} \cap \Omega_2)^C \cap (\Omega_{1a} \cap \Omega_3)^C \cap (\Omega_{1a} \cap \Omega_2)^C \cap \Omega_2 \\ &= \Omega_1^C \cap \Omega_2 \cap (\Omega_{1b} \cap \Omega_2)^C \cap (\Omega_{1a} \cap \Omega_3)^C \cap (\Omega_{1a} \cap \Omega_2)^C. \end{aligned} \quad (\text{I.35})$$

From (I.6),

$$R_1^C \cap R_2^C \cap R_3^C \cap R_4^C \cap R_5 = \Omega_1^C \cap \Omega_2 \cap (\Omega_{1a}^C \cup \Omega_3^C) \cap (\Omega_{1a}^C \cup \Omega_2^C) \cap (\Omega_{1b}^C \cup \Omega_2^C). \quad (\text{I.36})$$

From (6.97),

$$\begin{aligned} R_1^C \cap R_2^C \cap R_3^C \cap R_4^C \cap R_5 &= \Omega_1^C \cap (\Omega_2 \cap \Omega_{2b} \cap \Omega_3) \cap (\Omega_{1a}^C \cup \Omega_3^C) \cap (\Omega_{1a}^C \cup \Omega_2^C) \cap (\Omega_{1b}^C \cup \Omega_2^C) \\ &= ((\Omega_1^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3) \cap (\Omega_{1a}^C \cup \Omega_3^C)) \cap (\Omega_{1a}^C \cup \Omega_2^C) \cap (\Omega_{1b}^C \cup \Omega_2^C). \end{aligned} \quad (\text{I.37})$$

From (I.10),

$$\begin{aligned} R_1^C \cap R_2^C \cap R_3^C \cap R_4^C \cap R_5 &= ((\Omega_1^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3 \cap \Omega_{1a}^C) \cup (\Omega_1^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3 \cap \Omega_3^C)) \\ &\cap (\Omega_{1a}^C \cup \Omega_2^C) \cap (\Omega_{1b}^C \cup \Omega_2^C). \end{aligned} \quad (\text{I.38})$$

Since  $\Omega_3 \cap \Omega_3^C = \emptyset$ ,

$$\begin{aligned} R_1^C \cap R_2^C \cap R_3^C \cap R_4^C \cap R_5 &= ((\Omega_1^C \cap \Omega_{1a}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3) \cup \emptyset) \cap (\Omega_{1a}^C \cup \Omega_2^C) \cap (\Omega_{1b}^C \cup \Omega_2^C) \\ &= (\Omega_1^C \cap \Omega_{1a}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3 \cap (\Omega_{1a}^C \cup \Omega_2^C)) \cap (\Omega_{1b}^C \cup \Omega_2^C). \end{aligned} \quad (\text{I.39})$$

From (I.10),

$$\begin{aligned} R_1^C \cap R_2^C \cap R_3^C \cap R_4^C \cap R_5 \\ = ((\Omega_1^C \cap \Omega_{1a}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3 \cap \Omega_{1a}^C) \cup (\Omega_1^C \cap \Omega_{1a}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3 \cap \Omega_2^C)) \cap (\Omega_{1b}^C \cup \Omega_2^C). \end{aligned} \quad (\text{I.40})$$

Since  $\Omega_2 \cap \Omega_2^C = \emptyset$ ,

$$\begin{aligned} R_1^C \cap R_2^C \cap R_3^C \cap R_4^C \cap R_5 &= ((\Omega_1^C \cap \Omega_{1a}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3) \cup \emptyset) \cap (\Omega_{1b}^C \cup \Omega_2^C) \\ &= \Omega_1^C \cap \Omega_{1a}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3 \cap (\Omega_{1b}^C \cup \Omega_2^C). \end{aligned} \quad (\text{I.41})$$

From (I.10),

$$\begin{aligned} R_1^C \cap R_2^C \cap R_3^C \cap R_4^C \cap R_5 \\ = (\Omega_1^C \cap \Omega_{1a}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3 \cap \Omega_{1b}^C) \cup (\Omega_1^C \cap \Omega_{1a}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3 \cap \Omega_2^C). \end{aligned} \quad (\text{I.42})$$

Since  $\Omega_2 \cap \Omega_2^C = \emptyset$ ,

$$\begin{aligned} R_1^C \cap R_2^C \cap R_3^C \cap R_4^C \cap R_5 &= (\Omega_1^C \cap \Omega_{1a}^C \cap \Omega_{1b}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3) \cup \emptyset \\ &= \Omega_1^C \cap \Omega_{1a}^C \cap \Omega_{1b}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3. \end{aligned} \quad (\text{I.43})$$

Since  $(R_1^C \cap R_2^C \cap R_3^C \cap R_4^C \cap R_5) \cap R_2^C = R_1^C \cap R_2^C \cap R_3^C \cap R_4^C \cap R_5$ , from (6.95),

$$R_1^C \cap R_2^C \cap R_3^C \cap R_4^C \cap R_5 = (\Omega_1^C \cap \Omega_{1a}^C \cap \Omega_{1b}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3) \cap (\Omega_{1c} \cap \Omega_2)^C. \quad (\text{I.44})$$

From (I.6),

$$R_1^C \cap R_2^C \cap R_3^C \cap R_4^C \cap R_5 = (\Omega_1^C \cap \Omega_{1a}^C \cap \Omega_{1b}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3) \cap (\Omega_{1c}^C \cup \Omega_2^C). \quad (\text{I.45})$$

From (I.10),

$$\begin{aligned} R_1^C \cap R_2^C \cap R_3^C \cap R_4^C \cap R_5 \\ = (\Omega_1^C \cap \Omega_{1a}^C \cap \Omega_{1b}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3 \cap \Omega_{1c}^C) \cup (\Omega_1^C \cap \Omega_{1a}^C \cap \Omega_{1b}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3 \cap \Omega_2^C). \end{aligned} \quad (\text{I.46})$$

Since  $\Omega_2 \cap \Omega_2^C = \emptyset$ ,

$$\begin{aligned} R_1^C \cap R_2^C \cap R_3^C \cap R_4^C \cap R_5 &= (\Omega_1^C \cap \Omega_{1a}^C \cap \Omega_{1b}^C \cap \Omega_{1c}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3) \cup \emptyset \\ &= \Omega_1^C \cap \Omega_{1a}^C \cap \Omega_{1b}^C \cap \Omega_{1c}^C \cap \Omega_2 \cap \Omega_{2b} \cap \Omega_3. \end{aligned} \quad (\text{I.47})$$

**Region**  $R_1^C \cap R_2^C \cap R_3^C \cap R_4^C \cap R_5^C$

This is the region where only a single tree or network can be found. Since Network 1 and Network 2 can only be found in regions where other trees and networks can be found, neither of these trees will be found in this region. There will be five non-overlapping subregions in this region. They are

$$\begin{cases} S_{1a} &= \Omega_1^C \cap \Omega_{1a} \cap \Omega_{1b}^C \cap \Omega_{1c}^C \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3^C, \\ S_{1b} &= \Omega_1^C \cap \Omega_{1a}^C \cap \Omega_{1b} \cap \Omega_{1c}^C \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3^C, \\ S_{1c} &= \Omega_1^C \cap \Omega_{1a}^C \cap \Omega_{1b}^C \cap \Omega_{1c} \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3^C, \\ S_{2b} &= \Omega_1^C \cap \Omega_{1a}^C \cap \Omega_{1b}^C \cap \Omega_{1c}^C \cap \Omega_2^C \cap \Omega_{2b} \cap \Omega_3^C, \\ S_3 &= \Omega_1^C \cap \Omega_{1a}^C \cap \Omega_{1b}^C \cap \Omega_{1c}^C \cap \Omega_2^C \cap \Omega_{2b}^C \cap \Omega_3. \end{cases} \quad (\text{I.48})$$





# References

- Elizabeth S Allman and John A Rhodes. The identifiability of tree topology for phylogenetic models, including covarion and mixture models. *Journal of Computational Biology*, 13(5): 1101–1113, 2006.
- Elizabeth S Allman and John A Rhodes. Identifying evolutionary trees and substitution parameters for the general markov model with invariable sites. *Mathematical biosciences*, 211(1): 18–33, 2008.
- Elizabeth S Allman and John A Rhodes. The identifiability of covarion models in phylogenetics. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 6(1):76–88, 2009.
- Elizabeth S Allman, Cécile Ané, and John A Rhodes. Identifiability of a markovian model of molecular evolution with gamma-distributed rates. *Advances in Applied Probability*, pages 229–249, 2008.
- Elizabeth S Allman, Catherine Matias, and John A Rhodes. Identifiability of parameters in latent structure models with many observed variables. *The Annals of Statistics*, pages 3099–3132, 2009.
- Elizabeth S Allman, Catherine Matias, and John A Rhodes. Parameter identifiability in a class of random graph mixture models. *Journal of Statistical Planning and Inference*, 141(5):1719–1736, 2011a.
- Elizabeth S Allman, Sonia Petrovic, John A Rhodes, and Seth Sullivant. Identifiability of two-tree mixtures for group-based models. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 8(3):710–722, 2011b.
- Elizabeth S Allman, John A Rhodes, Elena Stanghellini, and Marco Valtorta. Parameter identifiability of discrete bayesian networks with hidden variables. *Journal of Causal Inference*, 3(2):189–205, 2015.
- Edgar Anderson and Leslie Hubricht. Hybridization in tradescantia. iii. the evidence for introgressive hybridization. *American Journal of Botany*, pages 396–402, 1938.
- Edgar Anderson et al. Introgressive hybridization. *Introgressive hybridization.*, 1949.

- GE Andersson, Olof Karlberg, Björn Canbäck, and Charles G Kurland. On the origin of mitochondria: a genomics perspective. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 358(1429):165–179, 2003.
- Vineet Bafna and Vikas Bansal. The number of recombination events in a sample history: conflict graph and lower bounds. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 1(2):78–90, 2004.
- Hans-Jürgen Bandelt and Andreas WM Dress. Split decomposition: a new and useful approach to phylogenetic analysis of distance data. *Molecular phylogenetics and evolution*, 1(3):242–252, 1992.
- Mihaela Baroni, Charles Semple, and Mike Steel. A framework for representing reticulate evolution. *Annals of Combinatorics*, 8(4):391–408, 2005.
- JD Bashford, PD Jarvis, JG Sumner, and MA Steel.  $U(1) \times U(1) \times U(1)$  symmetry of the Kimura 3ST model and phylogenetic branching processes. *Journal of Physics A: Mathematical and General*, 37(8):L81, 2004.
- David Bryant. Extending tree models to split networks. In *Algebraic statistics for computational biology*, pages 297–368. Cambridge University Press, 2005.
- David Bryant. Hadamard phylogenetic methods and the  $n$ -taxon process. *Bulletin of mathematical biology*, 71(2):339–351, 2009.
- O Peter Buneman. The recovery of trees from measures of dissimilarity. *Mathematics in the archaeological and historical sciences*, 1971.
- James A Cavender and Joseph Felsenstein. Invariants of phylogenies in a simple case with discrete states. *Journal of classification*, 4(1):57–71, 1987.
- Persi Diaconis. Group representations in probability and statistics. *Lecture Notes-Monograph Series*, pages i–192, 1988.
- Dan Gusfield, Satish Eddhu, and Charles Langley. Efficient reconstruction of phylogenetic networks with constrained recombination. In *Bioinformatics Conference, 2003. CSB 2003. Proceedings of the 2003 IEEE*, pages 363–374. IEEE, 2003.
- Dan Gusfield, Satish Eddhu, and Charles Langley. The fine structure of galls in phylogenetic networks. *INFORMS Journal on Computing*, 16(4):459–469, 2004.
- Jacques Hadamard. Résolution dune question relative aux déterminants. *Bull. sci. math*, 17(1):240–246, 1893.
- Brendan Hassett. *Introduction to algebraic geometry*. Cambridge University Press, 2007.

- S Blair Hedges, Julie Marin, Michael Suleski, Madeline Paymer, and Sudhir Kumar. Tree of life reveals clock-like speciation and diversification. *Molecular biology and evolution*, page msv037, 2015.
- Jotun Hein. Reconstructing evolution of sequences subject to recombination using parsimony. *Mathematical biosciences*, 98(2):185–200, 1990.
- Jotun Hein. A heuristic method to reconstruct the history of sequences subject to recombination. *Journal of Molecular Evolution*, 36(4):396–405, 1993.
- MD Hendy, David Penny, and MA Steel. A discrete fourier analysis for evolutionary trees. *Proceedings of the National Academy of Sciences*, 91(8):3339–3343, 1994.
- Michael D Hendy. The relationship between simple evolutionary tree models and observable sequence data. *Systematic Biology*, 38(4):310–321, 1989.
- Michael D Hendy. Hadamard conjugation: an analytic tool for phylogenetics. *Math. of Evolution and Phylogeny*, pages 143–177, 2005.
- Michael D Hendy and David Penny. A framework for the quantitative study of evolutionary trees. *Systematic Biology*, 38(4):297–309, 1989.
- Michael D Hendy and David Penny. Spectral analysis of phylogenetic data. *Journal of classification*, 10(1):5–24, 1993.
- Jody Hey. Isolation with migration models for more than two populations. *Molecular biology and evolution*, 27(4):905–920, 2010.
- Alan Hodgkinson and Adam Eyre-Walker. Human triallelic sites: evidence for a new mutational mechanism? *Genetics*, 184(1):233–241, 2010.
- Barbara R Holland, Hamish G Spencer, Trevor H Worthy, and Martyn Kennedy. Identifying cliques of convergent characters: concerted evolution in the cormorants and shags. *Systematic Biology*, page syq023, 2010.
- Richard R Hudson and Norman L Kaplan. Statistical properties of the number of recombination events in the history of a sample of dna sequences. *Genetics*, 111(1):147–164, 1985.
- Daniel H Huson, Tobias Klöpper, Pete J Lockhart, and Mike A Steel. Reconstruction of reticulate networks from gene trees. In *Research in Computational Molecular Biology*, pages 233–249. Springer, 2005.
- Daniel H Huson, Regula Rupp, and Celine Scornavacca. *Phylogenetic networks: concepts, algorithms and applications*. Cambridge University Press, 2010.
- Guohua Jin, Luay Nakhleh, Sagi Snir, and Tamir Tuller. Maximum likelihood of phylogenetic networks. *Bioinformatics*, 22(21):2604–2611, 2006.

- Thomas H Jukes and Charles R Cantor. Evolution of protein molecules. *Mammalian protein metabolism*, 3:21–132, 1969.
- Patrick J Keeling and Jeffrey D Palmer. Horizontal gene transfer in eukaryotic evolution. *Nature Reviews Genetics*, 9(8):605–618, 2008.
- Motoo Kimura. Estimation of evolutionary distances between homologous nucleotide sequences. *Proceedings of the National Academy of Sciences*, 78(1):454–458, 1981.
- James A Lake. A rate-independent technique for analysis of nucleic acid sequences: evolutionary parsimony. *Molecular Biology and Evolution*, 4(2):167–191, 1987.
- F-J Lapointe. How to account for reticulation events in phylogenetic analysis: a comparison of distance-based methods. *Journal of classification*, 17(2):175–184, 2000.
- Pierre Legendre and Vladimir Makarenkov. Reconstruction of biogeographic and evolutionary networks using reticulograms. *Systematic Biology*, 51(2):199–216, 2002.
- John Little, D Cox, and D OShea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra (Undergraduate Texts in Mathematics)*. Springer, New York, 1992.
- Wayne P Maddison. Gene trees in species trees. *Systematic biology*, 46(3):523–536, 1997.
- Frederick A Matsen. Fourier transform inequalities for phylogenetic trees. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 6(1):89–95, 2009.
- Simon R Myers and Robert C Griffiths. Bounds on the minimum number of recombination events in a sample history. *Genetics*, 163(1):375–394, 2003.
- Luay Nakhleh. Evolutionary phylogenetic networks: models and issues. In *Problem solving handbook in computational biology and bioinformatics*, pages 125–158. Springer, 2011.
- Luay Nakhleh, Tandy Warnow, and C Randal Linder. Reconstructing reticulate evolution in species: theory and practice. In *Proceedings of the eighth annual international conference on Research in computational molecular biology*, pages 337–346. ACM, 2004.
- Joe Parker, Georgia Tsagkogeorga, James A Cotton, Yuan Liu, Paolo Provero, Elia Stupka, and Stephen J Rossiter. Genome-wide signatures of convergent evolution in echolocating mammals. *Nature*, 502(7470):228–231, 2013.
- David Posada and Keith A Crandall. Intraspecific gene genealogies: trees grafting into networks. *Trends in Ecology & Evolution*, 16(1):37–45, 2001.
- OLE Seehausen, Gaku Takimoto, Denis Roy, and Jukka Jokela. Speciation reversal and biodiversity dynamics with hybridization in changing environments. *Molecular Ecology*, 17(1):30–44, 2008.

- Charles Semple and Mike A Steel. *Phylogenetics*, volume 24. Oxford University Press, 2003.
- Samuel K Sheppard, Noel D McCarthy, Daniel Falush, and Martin CJ Maiden. Convergence of campylobacter species: implications for bacterial evolution. *Science*, 320(5873):237–239, 2008.
- Samuel K Sheppard, Noel D McCarthy, Keith A Jolley, and Martin CJ Maiden. Introgression in the genus campylobacter: generation and spread of mosaic alleles. *Microbiology*, 157(4):1066–1074, 2011.
- JJ Sylvester. Thoughts on inverse orthogonal matrices, simultaneous sign successions, and tessellated pavements in two or more colours, with applications to newtons rule, ornamental tile-work, and the theory of numbers. *Philosophical Magazine*, 34:461–475, 1867.
- Yun S Song and Jotun Hein. On the minimum number of recombination events in the evolutionary history of dna sequences. *Journal of Mathematical Biology*, 48(2):160–186, 2004.
- MA Steel, MD Hendy, LA Székely, and PL Erdős. Spectral analysis and a closest tree method for genetic sequences. *Applied mathematics letters*, 5(6):63–67, 1992.
- Mike Steel. Should phylogenetic models be trying to fit an elephant? *Trends in Genetics*, 21(6):307–309, 2005.
- Bernd Sturmfels. Ideals, varieties and *Macaulay 2*. In *Computations in algebraic geometry with Macaulay 2*, pages 3–15. Springer, 2002.
- Jeremy G Sumner, Peter D Jarvis, Jesús Fernández-Sánchez, Bodie T Kaine, Michael D Woodhams, and Barbara R Holland. Is the general time-reversible model bad for molecular phylogenetics? *Systematic biology*, page sys042, 2012a.
- Jeremy G Sumner, Peter D Jarvis, and Barbara R Holland. A tensorial approach to the inversion of group-based phylogenetic models. *BMC evolutionary biology*, 14(1):236, 2014.
- JG Sumner, Jesús Fernández-Sánchez, and PD Jarvis. Lie markov models. *Journal of theoretical biology*, 298:16–31, 2012b.
- JG Sumner, BR Holland, and PD Jarvis. The algebra of the general markov model on phylogenetic trees and networks. *Bulletin of mathematical biology*, 74(4):858–880, 2012c.
- EB Taylor, JW Boughman, M Groenenboom, M Sniatynski, D Schluter, and JL Gow. Speciation in reverse: morphological and genetic evidence of the collapse of a three-spined stickleback (*Gasterosteus aculeatus*) species pair. *Molecular Ecology*, 15(2):343–355, 2006.
- Günter Theißen. Orthology: secret life of genes. *Nature*, 415(6873):741–741, 2002.
- Arndt von Haeseler and Gary A Churchill. Network models for sequence evolution. *Journal of molecular evolution*, 37(1):77–85, 1993.

Lusheng Wang, Kaizhong Zhang, and Louxin Zhang. Perfect phylogenetic networks with recombination. *Journal of Computational Biology*, 8(1):69–78, 2001.

Don E Wilson and DeeAnn M Reeder. *Mammal species of the world: a taxonomic and geographic reference*, volume 1. JHU Press, 2005.